



Titre: Architecture et algorithmes de routage pour un réseau ad hoc
Title: dense

Auteur: Fabien Nimbona
Author:

Date: 2004

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Nimbona, F. (2004). Architecture et algorithmes de routage pour un réseau ad hoc
Citation: dense [Master's thesis, École Polytechnique de Montréal]. PolyPublie.
<https://publications.polymtl.ca/7198/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/7198/>
PolyPublie URL:

**Directeurs de
recherche:**
Advisors:

Programme: Unspecified
Program:

UNIVERSITÉ DE MONTRÉAL

ARCHITECTURE ET ALGORITHMES DE ROUTAGE POUR
UN RÉSEAU AD HOC DENSE

FABIEN NIMBONA
DÉPARTEMENT DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLOME DE MAÎTRISE EN SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)

AVRIL 2004



National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services

Acquisitions et
services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-91958-7

Our file Notre référence

ISBN: 0-612-91958-7

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this dissertation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de ce manuscrit.

While these forms may be included in the document page count, their removal does not represent any loss of content from the dissertation.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

ARCHITECTURE ET ALGORITHMES DE ROUTAGE POUR
UN RÉSEAU AD HOC DENSE

présenté par : NIMBONA Fabien

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. GALINIER Philippe, Doct., président

M. PIERRE Samuel, Ph.D., membre et directeur de recherche

M. GUIBAUT François, Ph.D., membre

A Laetitia,
A Fiorelle-Leccia,

REMERCIEMENTS

À la fin de ce travail, qu'il me soit permis de remercier les personnes qui ont contribué à sa réalisation. Mes remerciements sont adressés principalement au professeur Samuel PIERRE pour avoir accepté la direction de ma recherche et financé ma dernière session. Ses conseils, ses orientations et ses critiques, malgré ses multiples obligations, m'ont été d'une grande importance.

Je remercie également l'Agence Canadienne de Développement International (ACDI) à travers son programme canadien des bourses de la francophonie (PCBF) pour avoir financé mes études de maîtrise.

Je souhaite aussi remercier tous les membres du LARIM pour le climat de travail et leur entraide quotidienne.

Que les membres de ma famille, plus particulièrement ma femme, mes parents, mes frères et ma sœur trouvent, à travers ce travail, une satisfaction de leur soutien moral et matériel de tous les jours.

Enfin, à toute autre personne qui, de près ou de loin, a contribué à la réussite de ce travail, je dis grand merci.

RÉSUMÉ

Les réseaux mobiles ad hoc constituent une alternative intéressante pour les endroits qui, malgré l'installation à grande échelle des réseaux de télécommunications (réseau fixe, réseau cellulaire classique et Internet), n'ont toujours pas bénéficié d'infrastructures dont la mise en place peut s'avérer peu rentable, temporaire ou quasi-impossible. Cependant, les réseaux mobiles ad hoc ont introduit un certain nombre de problèmes qui constituent des défis de recherche. En effet, leurs caractéristiques font qu'ils utilisent, dans le routage, des mécanismes de diffusion alors que l'un des défis dans les réseaux sans fil est l'optimisation de la bande passante. Ce mémoire propose une architecture, un algorithme de formation de grappes ainsi que des algorithmes de routage pour garantir une qualité de service dans les réseaux mobiles ad hoc denses.

L'architecture que nous proposons est constituée de grappes. Il y a, dans chaque grappe, une tête de grappe qui gère toutes communications intra-grappes. Cependant, la tête de grappe constitue un point critique du réseau. Ainsi, nous prévoyons une tête de grappe de secours qui prend automatiquement la relève dès que la principale tombe en panne. En outre, comme le réseau considéré est dense, les passerelles et les passerelles réparties peuvent être dans une même zone de portée de transmission mutuelle. Nous proposons de construire une maille entre ces dernières afin d'améliorer la fiabilité des communications inter-grappes.

Dans la littérature, le protocole *multicast* ODMRP est identifié comme étant le plus robuste et le plus adapté au routage dans les réseaux mobiles ad hoc. Une méthode de prédiction de la mobilité a été déjà intégrée à ce protocole pour choisir une route sur la base de la stabilité au lieu du délai. Cependant, une route peut être stable et imposer un délai inacceptable pour les communications temps réel. Ainsi, notre contribution vise à introduire des contraintes de délai, de capacité au niveau des nœuds intermédiaires et de bande passante. L'évaluation des performances s'est faite à l'aide du simulateur GloMoSim. Les résultats montrent que la détérioration des performances imposée par l'ajout des contraintes à cette version du protocole ODMRP est insignifiante.

L'algorithme d'élection que nous proposons (SRCAC) se base sur les caractéristiques physiques des éléments du réseau et du type de trafic pour déterminer un indice de potentialité. En outre, il se base sur les informations de localisation fournies par le système GPS pour déterminer le temps d'expiration d'un lien (LET : Link Expiration Time) entre une unité mobile et chaque nœud de son voisinage. La moyenne des LET, notée CET (Cluster Expiration Time), est l'indice de stabilité de la grappe qui aurait comme tête de grappe cette unité mobile. Nous définissons un indice d'élection qui est une combinaison linéaire de l'indice de potentialité et de l'indice de stabilité. Notre algorithme choisit comme tête de grappe, l'unité mobile qui a l'indice d'élection le plus grand. Nous l'avons comparé à un algorithme nommé WCA sur la base de la métrique du nombre de têtes de grappe.

Ainsi, nous avons généré aléatoirement 3 réseaux respectivement de 12, 15 et 20 unités mobiles. Les variables de la simulation utilisées pour valider l'implémentation et simuler l'algorithme WCA sont tirées de (Chatterjee et al., 2002). Nous avons utilisé 0.95 et 0.05 comme coefficients associés respectivement à l'indice de potentialité et à l'indice de stabilité. Les résultats montrent qu'à chaque fois les têtes de grappes choisis sont les plus pourvues en ressources. Ce qui n'est pas le cas pour l'algorithme WCA. La comparaison des deux algorithmes montre que le SRCAC est 4.9 % meilleur que le WCA.

ABSTRACT

In spite of the installation on a large scale of the telecommunications networks (fixed network, traditional cellular network and Internet network), some areas still have not infra-structure. Indeed, their installations are temporarily, impossible or not profitable. The Mobile Ad hoc Network (MANET) constitute an alternative and interesting solution for those places. However, they introduced some problems which are challenges for the research. Indeed, due to their characteristics, they use broadcast mechanism in routing protocols whereas one of the wireless network challenges is the band-width optimisation. In this work, we propose an architecture, a clustering algorithm as well as routing algorithms to guarantee quality of service in the dense mobile ad hoc networks.

The proposed architecture consists of clusters. In each cluster, a clusterhead manages all intra-clusters communications. However, the clusterhead is the bottleneck of the network. Thus, we propose a backup clusterhead which could become automatically the principal clusterhead when the first breaks down. Moreover, as the considered network is dense, the gateways and the distributed gateways can be in mutual transmission zone. In that case, it is better to build a mesh between them to improve the reliability of inter-clusters communications. In the literature, the ODMRP protocol has been identified as the most robust and adapted routing protocol to the mobile ad hoc network. The mobility prediction method was already integrated into this protocol to choose a route using the stability metric instead of the delay metric. However, a stable route can have a delay which can not be acceptable. Then, we introduce time, capacity and band-width constraints on the intermediate nodes. GloMoSim simulator is used to evaluate the performances. The results show that the deterioration of the performances imposed by the constraints on this ODMRP version is unimportant.

The proposed election algorithm is based on the physical characteristics of the network elements and the traffic type to determine the potentiality index. Moreover, it uses the GPS information to determine the link expiration time (LET) between a node and each element of its neighborhood. The average of the LET, noted CET (Cluster Expiration Time) is the stability index of the cluster which would have like

clusterhead this node. We define an election index as a linear combination of potentiality and stability indexes. Our algorithm chooses like clusterhead the node with the highest election index. We compared the proposed algorithm to WCA algorithm using clusterhead metric.

Thus, we generated by chance 3 networks respectively of 12, 15 and 20 nodes. To validate the implementation and the simulation of WCA algorithm, we use the variables from (Chatterjee and Al, 2002). For the SRCAC, we used 0.95 and 0.05 as coefficients associated to potentiality index and stability index. The results show that the selected clusterheads provided in resources. What is not the same case for the WCA algorithm. The comparison of the two algorithms shows that the SRCAC is 4.9 % better than the WCA.

TABLE DES MATIÈRES

DÉDICACE	iv
REMERCIEMENTS	v
RÉSUMÉ	vi
ABSTRACT.....	viii
TABLE DES MATIÈRES.....	x
LISTE DES TABLEAUX.....	xiii
LISTE DES FIGURES.....	xiv
LISTE DES SIGLES ET DES ABBRÉVIATIONS	xvi
CHAPITRE I : INTRODUCTION	1
1.1 Définitions et concepts de base.....	1
1.2 Éléments de la problématique	4
1.3 Objectifs de recherche	6
1.4 Esquisse méthodologique	7
1.5 Plan du mémoire	7
CHAPITRE II : PROBLÉMATIQUE DU ROUTAGE DANS LES	
RÉSEAUX AD HOC	8
2.1 Caractéristiques de base des réseaux mobiles ad hoc	8
2.2 Routage dans les réseaux mobiles ad hoc	9
2.2.1 Les protocoles de routage proactifs.....	10
2.2.1.1 Le protocole de routage HSR.....	11
2.2.1.2 Le protocole CGSR.....	12
2.2.2 Les protocoles de routage réactifs.....	14
2.2.2.1 Le protocole de routage DSR.....	15
2.2.2.2 Le protocole de routage AODV	17
2.2.2.3 Le protocole de routage CBRP	19
2.2.3 Limitations des protocoles de routage dans les réseaux ad hoc	21
2.2.4 Routage par la méthode de prédiction de la mobilité.....	22
2.2.4.1 Temps d'expiration d'un lien et d'une route	23
2.2.5 Routage multicast dans les réseaux ad hoc	24
2.2.6 Description de ODMRP	26

2.3 Algorithmes de subdivision en grappes	29
2.3.1 La mise en grappe	30
2.3.2 Quelques algorithmes d'élection des têtes de grappes	31
2.3.3 Brève description de l'algorithme	33
2.4 Architecture de base	35
2.4.1 Brève description du protocole ARC	36
2.4.2 Étape d'initialisation.....	36
2.4.3 Découverte de la topologie.....	37
CHAPITRE III : ARCHITECTURE ET ALGORITHMES DE	
ROUTAGE PROPOSÉS.....	39
3.1 Principe de l'architecture proposée et hypothèses sous-jacentes	39
3.1.1 Idée de base de l'architecture	40
3.1.2 Hypothèses	41
3.2 Présentation générale de notre architecture	42
3.2.1 Modèle simplifié de notre architecture	42
3.2.2 Initialisation	43
3.2.3 Aspects de fiabilité et de disponibilité du service	44
3.2.4 Communication entre le CLHP et le clhs.....	47
3.3 Algorithmes d'élection de tête de grappe	48
3.3.1 Philosophie de conception.....	48
3.3.2 Les bases de l'algorithme	53
3.3.3 Algorithme d'élection d'une tête de grappe	54
3.4 Routage inter-grappes	56
3.4.1 Choix d'un protocole adapté au routage inter-grappes	57
3.4.2 Application du mécanisme de prédiction de la mobilité avec contraintes	58
3.4.3 Détermination de la meilleure route.....	60
3.4.4 Exemple illustrant le routage inter-grappes	62
CHAPITRE IV: IMPLÉMENTATION ET ANALYSE DES RÉSULTATS	67
4.1 Implémentation de l'algorithme d'élection proposé	67
4.1.1 Principe de fonctionnement et paramètres de base	67
4.1.2 Détermination de l'indice de potentialité	69

4.1.3 Détermination du CET	70
4.1.4 Les classes utilisées pour implémenter notre algorithme	70
4.2 Évaluation des performances	71
4.3 Évaluation des algorithmes et des résultats	72
4.4 Implémentation du protocole ODMRP avec prédiction de la mobilité	80
4.4.1 Bref aperçu de Glomosim	80
4.4.2 ODMRP avec prédiction de la mobilité	82
4.4.3 ODMRP avec prédiction de la mobilité et contraintes des qualité de service	83
4.5 Évaluation des performances du protocole ODMRP	84
CHAPITRE V : CONCLUSION	87
5.1 Synthèse des travaux	87
5.2 Limitations des travaux	89
5.3 Travaux futurs	90
BIBLIOGRAPHIE	91

LISTE DES TABLEAUX

Tableau 3.1 Caractéristiques des liens de la Figure 3.10	65
Tableau 4.1 Coefficients d'indice de potentialité en fonction du type de trafic	70
Tableau 4.2 La mobilité et le degré de chacun des 15 éléments du réseau.....	75
Tableau 4.3 Comportement du « WCA » en « highest degree heuristic »	76
Tableau 4.4 Comportement du « WCA » en « Lowest-Id heuristic ».....	76
Tableau 4.5 Les indices de potentialité, de stabilité et d'élection	77
Tableau 4.6 Grappes formées à partir du réseau de la Figure 4.2.....	79

LISTE DES FIGURES

Figure 1.1 Modèle de réseaux mobiles avec infrastructures	2
Figure 1.2 Portée des unités mobiles en fonctionnement omnidirectionnel	3
Figure 2.1 Partitionnement du réseau en groupes	13
Figure 2.2 Illustration du protocole CGSR	14
Figure 2.3 Construction de l'enregistrement de route	16
Figure 2.4 Renvoi de la route	17
Figure 2.5 Diffusion du paquet Route Request (RREQ).....	19
Figure 2.6 Renvoi du paquet de réponse (RREP).....	19
Figure 2.7 Organisation du réseau dans le CBRP	20
Figure 2.8 Flux de requête (Flow-REQ)	24
Figure 2.9 Flux de sélection de route (Flow-SETUP).....	24
Figure 2.10 Illustration des protocoles basés sur l'approche d'arbre	26
Figure 2.11 Illustration des protocoles basés sur l'approche de maille.....	26
Figure 2.12 Illustration d'une maille	27
Figure 2.13 Concept de groupe de multi-diffusion	28
Figure 3.1 Modèle simplifié de l'architecture proposée.....	43
Figure 3.2 Procédure d'initialisation	46
Figure 3.3 Procédure de choix du type de mise à jour	49
Figure 3.4 Procédure de mise à jour locale	50
Figure 3.5 Procédure de mise à jour globale.....	51
Figure 3.6 Algorithme d'élection d'une tête de grappe	56
Figure 3.7 Procédure de détermination du type de trafic	59
Figure 3.8 Algorithme de routage inter-grappes	61
Figure 3.9 Procédure de détermination de la meilleure route	62
Figure 3.10 Méthode de prédiction de la mobilité avec contraintes	64
Figure 3.11 Illustration du modèle après suppression des nœuds et des liens ne respectant pas les contraintes.....	66
Figure 4.1 Diagramme des classes	71
Figure 4.2 Configuration du réseau de test	73

Figure 4.3 Comparaison des deux algorithmes sur un réseau de 12 éléments	76
Figure 4.4 Comparaison des deux algorithmes sur un réseau de 15 éléments	77
Figure 4.5 Nombre de têtes de grappes pour un réseau de 20 éléments ayant des portées de transmission différentes	78
Figure 4.6 Comparaison sur un réseau de 20 éléments ayant une même portée de transmission.....	78
Figure 4.7 Délai de bout-en-bout en fonction de la taille du groupe <i>multicast</i>	85
Figure 4.8 Surcharge de contrôle en fonction du nombre de sources	86

LISTE DES SIGLES ET DES ABRÉVIATIONS

ABR	Associativity-Based Routing
ALP	Adaptive Link state Protocol
AMRIS	Ad Hoc Multicast Routing Protocol using Increasing id-numberS
AMRoute	Ad hoc <i>Multicast</i> Routing
AODV	Ad hoc On-Demand Distance Vector
ARC	Adaptive Routing using Clusters
BS	Base Station (Station de base)
CAMP	Core-Assisted Mesh Protocol
CET	Cluster Expiration Time
CBRP	Cluster Based Routing Protocol
CGSR	Clustehead Gateway Switch Routing
CLHP	Clusterhead principal
CLHS	Clusterhead secondaire
(DG)	passerelle répartie (Distributed Gateway)
DREAM	Distance Routing Effect Algorithm for Mobility
DRDT	(Directionnal Receiver Directionnal Transmitter (Réception Directionnelle et Transmission Directionnelle))
DSDV	Destination Sequence Distance Vector
DSR	Dynamic Source Routing
FGMP	Forwarding Group Multicast Protocol
FSR	Fishe State Routing
G	Passerelle (Gateway)
GSR	Global State Routing
GPS	Global Positionning System
HLR	Home Location Register
HSR	Hierarchical State Routing
indice_pot	Indice de potentialité

IETF	Internet Engineering Task Force
INRIA	Institut National de Recherche en Informatique et en Automatique
LAR	Location-Aided Routing
LET	Link Expiration Time
LCC	Least Cluster Change
LMS	Location Management Server
LRR	Least Routing Protocols
MANET	Mobile Ad hoc Network
NTP	Network Time Protocol
ODMRP	On-Demand Multicast Routing Protocol
OROT	Omnidirectionnal Receiver Omnidirectionnal Transmitter (Réception Omnidirectionnelle et Transmission Omnidirectionnelle)
OLSR	Optimized Link State Routing
PCC	Plus Court Chemin
RDMAR	Relative Distance Micro-discovery Ad Hoc Routing
RET	Route Expiration Time
SRCAC	Stability and Resources Consideration Algorithm in Clustering ad hoc network
STAR	Source Tree Adaptive Routing
TORA	Temporary Ordered Routing Algorithm
T_c	Temps critique
T_d	Temps de transmission du dernier paquet
UM	Unité mobile
WCA	A Weighted Clustering Algorithm for Mobile Ad hoc Networks
WRP	Wireless Routing Protocol
ZRP	Zone Routing Protocol

CHAPITRE I

INTRODUCTION

La fin du vingtième siècle a été caractérisée par une avancée spectaculaire des nouvelles technologies de l'information et de la communication. Le réseau fixe s'est vu complété par le réseau cellulaire pour répondre à la mobilité de l'utilisateur. Cependant, le profit étant la principale motivation de l'investisseur, certaines régions n'ont pas bénéficié d'infrastructures de réseaux classiques parce que la mise en place de celles-ci était peu rentable, temporaire ou tout simplement quasi-impossible. Ces régions ont été laissées à un autre type de réseau mobile qualifié de *ad hoc*. Malgré leurs multiples applications, les *réseaux mobiles ad hoc* ont introduit un certain nombre de problèmes qui constituent autant de défis pour les chercheurs et qui inspirent ce mémoire. Dans ce premier chapitre d'introduction, nous précisons d'abord quelques définitions et concepts de base, puis nous explicitons les éléments de la problématique, formulons par la suite nos objectifs de recherche et la méthodologie pour les atteindre, pour terminer avec un énoncé des grandes lignes de ce mémoire.

1.1 Définitions et concepts de base

En réponse à la volonté de l'utilisateur d'accéder à l'information indépendamment de sa localisation géographique, on a introduit les *réseaux mobiles* ou sans fil. La plupart de ces réseaux utilisent le modèle cellulaire, comme le montre la Figure 1.1. Dans ce modèle, on retrouve des sites fixes appelés *stations de base (BS)*. Une station de base est dotée d'une interface sans fil lui permettant de communiquer directement avec toute autre unité mobile (UM) localisée dans une zone géographique nommée *cellule*. Les stations de base sont interconnectées par un réseau de communication filaire. Cette catégorie constitue l'ensemble des *réseaux mobiles avec infrastructure*. Cependant, il en existe une autre qui n'a aucune infrastructure préexistante. On l'appelle *réseau mobile ad hoc*.

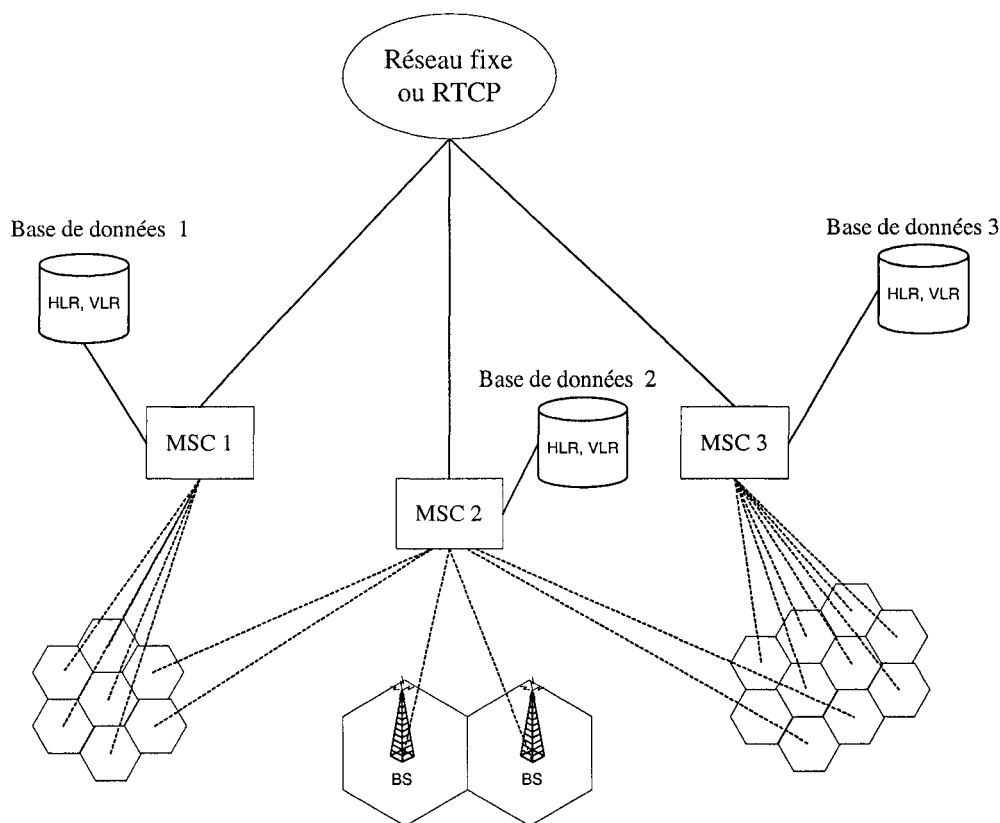


Figure 1.1 Modèle de réseaux mobiles avec infrastructures

Un *réseau mobile ad hoc*, connu généralement sous le nom de MANET (Mobile Ad hoc Network), est un réseau à topologie dynamique, formé d'un ensemble d'unités mobiles se déplaçant librement et interconnectées grâce à une technologie de communication radio, sans l'aide d'infrastructures ni d'administration centralisées. Ces unités mobiles sont équipées de cartes d'interface radio et de couches de protocoles adaptées aux fins de communication. L'interface radio est dotée d'antennes pouvant émettre dans toutes les directions ou dans une direction bien précise; on parle respectivement de fonctionnement *omnidirectionnel* et *directionnel*. Dans un réseau quelconque, on désigne par *route* ou *chemin* la séquence de liaisons qui permet de relier une source à une destination donnée. Le *roulage* est une opération par laquelle une information est acheminée d'une source à une destination en choisissant, sur la base d'une métrique donnée, un chemin parmi plusieurs.

Dans les réseaux cellulaires, le médium de communication peut être réutilisé sans entraîner de collisions. On parle alors de *réutilisation spatiale*. Cependant, dans les *réseaux mobiles ad hoc*, certaines paires de stations peuvent être hors de portée mutuelle, comme l'illustre la Figure 1.2. D'où le concept de *roulage interne* par les nœuds intermédiaires.

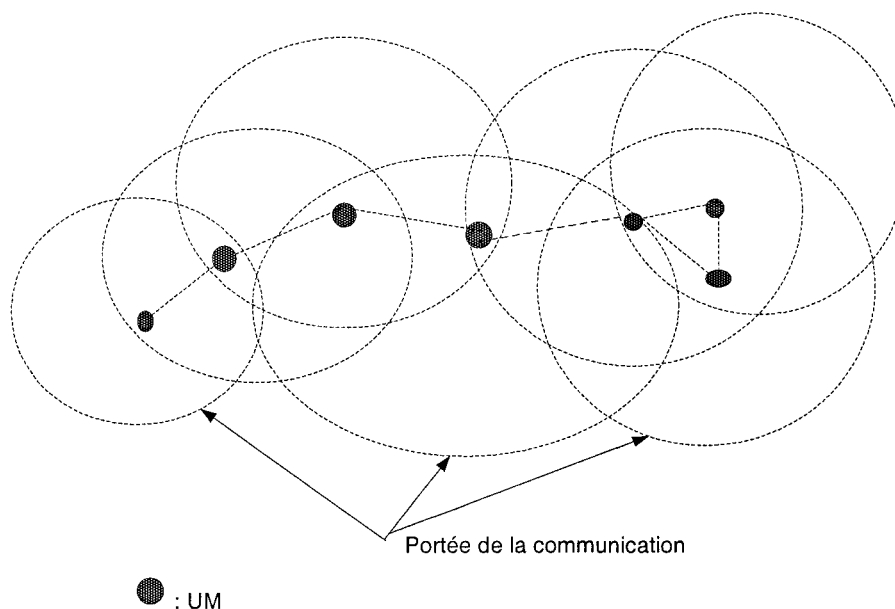


Figure 1.2 Portée des unités mobiles en fonctionnement omnidirectionnel

Si deux stations émettent simultanément vers une même unité mobile, elles vont générer des interférences et le problème est connu sous le nom de *stations cachées*. Les routes sont établies soit à l'avance ou à la demande par la source. On parle respectivement de *roulage proactif* et *réactif*.

Quand le réseau est dense, il est généralement subdivisé, pour son étude, en petits groupes appelés *grappes* (clusters) et, dans chaque grappe, un coordonnateur nommé *tête de grappe* (*clusterhead*) est élu. Une unité mobile (UM) peut être en communication directe avec plus d'une *tête de grappe*, on l'appelle *passerelle* (*Gateway*). On désigne par *passerelle répartie* (*Distributed Gateway*) toute paire d'unités mobiles qui, malgré qu'elle soit dans deux grappes différentes, peut communiquer directement. Dans ce réseau, certaines applications nécessitent des

communications un à un ou un à plusieurs, on les qualifie respectivement de communications «unicast» (*diffusion sélective*) et «multicast» (*multi-diffusion*). Le concept de *groupe de multi-diffusion* a été introduit pour supporter ces dernières. Ainsi, un émetteur considère l'ensemble des destinations comme une entité unique.

1.2 Éléments de la problématique

La portée des unités mobiles étant limitée, il se peut qu'une source et une (ou plusieurs) destination ne soient pas en communication directe. D'où l'intervention du routage par les nœuds de transit. Parmi les protocoles de routage proposés par l'IETF (Internet Engineering Task Force), on retrouve deux groupes principaux : les protocoles proactifs et les protocoles réactifs. La majorité de ceux qui se retrouvent dans les deux groupes utilise les messages de signalisation qui sont soit des échanges de tables de routage soit des requêtes de construction, de maintenance et de mise à jour des routes entre une origine et une destination. Ces messages sont souvent transmis par diffusion (broadcast), ce qui génère un trafic de contrôle excessif consommant ainsi une bande passante déjà réduite. En outre, pour les protocoles réactifs, le fait que la route soit établie à la demande introduit un délai supplémentaire, ce qui constitue une limitation pour les applications temps réel.

Avec l'avènement du système GPS (Global Positionning System) (Kaplan, 1996), des protocoles basés sur la localisation ont vu le jour (Young-Bae et Vaidya, 1998; Young-Bae et Vaidya, 2000). Actuellement, une méthode de prédiction de la mobilité (Su et al., 1999) utilise les éléments de localisation fournis par ce système pour estimer le temps d'expiration d'un lien (LET: Link Expiration Time) entre deux nœuds adjacents et par la suite déduire celui d'un chemin (une route), le RET (Route Expiration Time). Malgré que cette méthode optimise la bande passante, la route choisie peut imposer un délai inacceptable pour les applications temps réel. En outre, elle peut vouloir router plus de trafic qu'un lien n'est capable d'en accepter.

Quand le réseau est dense, la notion de mise en grappe (clustering) est invoquée. Son objectif principal est la réduction du nombre de participants au mécanisme de

diffusion. En effet, c'est un mécanisme qui consomme les ressources du réseau et la pénalité augmente avec la taille de ce dernier. Plusieurs mécanismes de mise en grappe ont été proposés dans la littérature (Vaidya et al., 1997; Lin et Gerla, 1997). Tous supposent une collecte des informations des voisins d'un nœud, ce qui consomme la bande passante. Certains travaux de recherche déterminent des protocoles répartis, d'autres se sont orientés vers la détermination d'un ensemble dominant d'unités mobiles. Cet ensemble contient des *têtes de grappe* qui sont responsables des communications intra-grappes et servent de point de routage pour les communications inter-grappes.

Les premiers algorithmes d'élection de têtes de grappe se basaient sur des paramètres comme le numéro d'identification (ID), le degré (nombre de voisins directs) d'un nœud et la capacité d'agir comme tête de grappe (Gerla et Tsai, 1995). Peu de temps après, d'autres travaux ont montré que les algorithmes basés sur un de ces paramètres avaient des limitations et ne s'appliquaient qu'à un cas particulier de réseaux sans fil. En effet, une tête de grappe peut avoir plus de voisins qu'elle ne peut supporter à cause des limitations des ressources. Avoir plus de têtes de grappe entraîne un bon débit de transfert mais introduit une latence. D'un autre côté l'attribution des ID est arbitraire et ne tient pas compte de la possibilité de devenir une tête de grappe. Le choix basé sur la capacité d'agir comme tête de grappe ne tient pas compte du temps d'exécution de l'algorithme, alors que c'est un élément important dans l'estimation du délai moyen de transmission. À partir de ces limitations, un autre algorithme a été proposé (Chatterjee et al., 2002) et optimise les paramètres comme le nombre de voisins qu'une tête de grappe peut supporter, la puissance de transmission, la mobilité et la puissance des batteries des unités mobiles. Dans tous ces algorithmes, il ressort que, dans une grappe, les nœuds communiquent entre eux à travers la tête de grappe. Cette dernière devient un nœud central et constitue un point critique de tout le réseau. En effet, quand ce nœud tombe en panne, toutes les informations relatives à la grappe sont perdues. En outre, la disponibilité du service est anéantie durant le temps d'exécution de l'algorithme d'élection d'une nouvelle *tête de grappe*, ce qui a une influence négative pour les applications temps réel.

L'utilisation d'une source d'énergie durable et des antennes directionnelles peut augmenter la portée de transmission et minimiser par la suite le nombre de sauts qui demeure un facteur important dans le calcul du délai moyen de transmission. Le routage inter-grappes a fait également l'objet de recherche. Certains travaux (Chatterjee et al., 2002) supposent que deux têtes de grappes peuvent communiquer directement et indiquent que la consommation d'énergie liée à cette communication est grande par rapport à celle nécessaire aux communications inter-grappes. D'autres algorithmes proposent soit un routage hiérarchique, soit un routage réactif (McDonald et Znati, 1999). Actuellement, un routage adaptatif utilisant la notion de grappes a été combiné avec un protocole réactif à la demande nommé AODV (Access On-Demand Distance Vector) (Royer, 2000) qui, en plus des problèmes de protocoles réactifs, entraîne une perte de paquets en cas de collisions ou avant la notification d'une défaillance d'un lien.

1.3 Objectifs de recherche

Ce mémoire a pour objectif principal de proposer une architecture globale et un algorithme de routage *multicast* utilisant la méthode de prédiction de la mobilité soumis aux contraintes de délai et de capacité pour un réseau ad hoc dense. De manière plus spécifique, il vise à:

- proposer des schémas de réplication de têtes de grappe pour améliorer la disponibilité du service dans les réseaux ad hoc ;
- concevoir un algorithme de formation des grappes en s'inspirant des schémas de réplication proposés pour optimiser le délai de transmission ;
- inclure à l'un des protocoles de routage *multicast* robuste une méthode de prédiction de la mobilité ainsi que les contraintes de délai et de capacité pour les communications inter-grappes ;
- évaluer la performance de l'architecture et des différents algorithmes proposés en regard des meilleurs algorithmes recensés dans la littérature.

1.4 Esquisse méthodologique

La conception d'un algorithme de formation des grappes suppose l'existence, en plus du système GPS, d'une source d'énergie durable et l'utilisation des antennes directionnelles. Les unités mobiles sont triées en fonction de leur potentialité à devenir des têtes de grappes (capacité de stockage, vitesse du processeur, mémoire RAM) ainsi que la stabilité qu'elles peuvent procurer aux grappes formées. Pour chaque unité mobile, nous faisons un changement de référentiel. Les autres unités mobiles obtiennent des coordonnées relatives. Nous reprenons certains paramètres comme la mobilité, le degré (qui varie avec le temps) pour choisir deux meilleures têtes de grappe. Parmi les schémas de réplication qui existent (réplication active, passive, semi-passive ou semi-active), nous choisissons le plus convenable à la mise en réplication des deux têtes de grappe (principale et secondaire). Pour les communications inter-grappes, nous construisons une maille entre les *passerelles* et les *passerelles réparties*. Un des meilleurs protocoles de routage "*multicast*" est choisi. Nous ajoutons à ce dernier les contraintes de qualité de service. La route sera déterminée en fonction de trois paramètres (RET, délai, capacité) pour les applications temps réel, et seulement le RET pour les applications du moindre effort. Des simulations seront effectuées pour évaluer les performances de nos propositions.

1.5 Plan du mémoire

Ce mémoire comprend cinq chapitres. À la suite du premier chapitre d'introduction, le second chapitre présente une revue des travaux effectués sur le routage en général et le routage *multicast* avec la méthode de la prédiction de la mobilité en particulier. Ce même chapitre analyse les algorithmes d'élection de têtes de grappe appliqués jusqu'à date. Le troisième chapitre décrit l'architecture proposée ainsi que les algorithmes d'élection et de routage retenus pour les communications inter-grappes. Le chapitre 4 présente les détails d'implémentation des algorithmes proposés et analyse les résultats. Dans le chapitre 5, une synthèse des travaux ainsi que des orientations pour des recherches futures sont présentées.

CHAPITRE II

PROBLÉMATIQUE DU ROUTAGE DANS LES RÉSEAUX AD HOC

Dans les réseaux ad hoc, il faut trouver un protocole de routage qui permet une utilisation optimale des ressources, en l'occurrence la bande passante et l'énergie. D'autre part, ce protocole doit minimiser le délai moyen de transmission entre une source et une destination tout en améliorant le débit de trafic transmis (throughput) par unité de temps. Ce problème est difficile et complexe surtout lorsque le réseau est dense. Ce chapitre analyse la problématique du routage dans les réseaux mobiles ad hoc. Après un aperçu sommaire des caractéristiques, des applications et de quelques protocoles de routage dans les réseaux mobiles ad hoc, nous décrivons quelques protocoles *unicast* et *multicast* avant de parler des algorithmes de mise en grappe et d'élection des têtes de grappes. Nous terminons par une brève description d'une architecture de base et d'un des protocoles utilisés par cette architecture.

2.1 Caractéristiques de base des réseaux mobiles ad hoc

Les réseaux mobiles ad hoc sont principalement caractérisés par une topologie dynamique, une absence d'infrastructures et des liaisons à débits variables. Leur bande passante, leur énergie et leur sécurité physique sont limitées.

Actuellement, les réseaux mobiles ad hoc sont utilisés dans les applications industrielles et commerciales incluant les échanges de données. La technologie MANET (Mobile Ad hoc Network), combinée avec une distribution de l'information basée sur les satellites, est une méthode extrêmement flexible pouvant permettre d'établir des communications destinées aux opérations de sauvetage ou d'autres scénarios d'urgence demandant un déploiement rapide de solutions (Corson et Macker, 1999). Ils intéressent d'un côté les applications militaires visant à mettre en place les réseaux de tactiques mobiles adaptés aux mouvements de troupes et d'un autre côté l'interconnexion des réseaux Ethernet lorsque la mise en place du câblage est impossible.

On peut également recourir à ces réseaux pour des besoins de communication entre personnes ou véhicules proches. Ainsi, un projet nommé SOLIDOR de l'INRIA (Institut National de Recherche en Informatique et en Automatique) en France a défini un concept de système d'information spontané nommé SIS (Banâtre et al., 2000) qui pourra servir à la conception de nouveaux systèmes anticollisions pour les véhicules où une commande de freinage génère automatiquement une information aux véhicules voisins. Ce système pourrait également être utilisé pour surveiller ou faire communiquer des robots mobiles. Chaque unité mobile peut communiquer avec n'importe quelle autre unité mobile de son groupe par une liaison radio et peut également accéder, via un point d'accès, aux ressources d'un réseau fixe.

Les applications des réseaux mobiles ad hoc s'avèrent nombreuses et nous nous permettons dans cette revue de citer certaines d'entre elles. Ces réseaux peuvent aussi être utilisés :

- pour accéder à distance à une information de réseau d'entreprise ou à une base de données ;
- dans des environnements difficiles à câbler ;
- dans une situation de déplacement fréquent ;
- dans les réseaux locaux temporaires, spécifiquement pour des projets spéciaux ou lors des heures de pointe ;
- dans des conférences, etc .

2.2 Routage dans les réseaux mobiles ad hoc

Comme la portée de transmission des unités mobiles est limitée, une source et une (ou plusieurs) destination(s) peuvent ne pas être en communication directe. D'où l'intervention du routage.

Rappelons que le routage est une opération par laquelle une information est acheminée d'une source à une destination en choisissant, sur la base d'une métrique donnée, un chemin parmi plusieurs. Un protocole de routage est une stratégie qui permet

d'établir des routes correctes et efficaces entre une source et une destination. Dans les réseaux mobiles ad hoc, les protocoles de routage sont souvent classés en deux grandes catégories : les protocoles de routage réactifs et les protocoles de routage proactifs.

2.2.1 Les protocoles de routage proactifs

Les protocoles de routage proactifs établissent et maintiennent les routes entre toute paire origine-destination de manière continue en se basant sur l'échange périodique des tables de routage. Quand une source veut transmettre des données, elle connaît à l'avance la route vers la destination. Les protocoles de routage proactifs sont basés sur une stratégie du plus court chemin qu'on retrouve dans les réseaux fixes. Ces protocoles utilisent deux méthodes principales, à savoir la méthode du *vecteur de distance* et la méthode de l'*état de lien*.

- La méthode du *vecteur de distance* (*distance vector*). C'est une méthode basée sur l'algorithme réparti de Bellman-Ford. Dans cette méthode, chaque unité mobile diffuse à ses voisins sa vision des distances qui le séparent des autres unités mobiles du réseau. L'utilisation de cette méthode se heurte à un problème nommé "*counting to infinity*". Ce dernier découle du fait que l'algorithme de Bellman-Ford ne dispose pas de mécanisme qui détermine à quel moment le réseau peut arrêter d'incrémenter la distance correspondant à une destination donnée.
- La méthode de l'*état de lien* (link state). Dans cette méthode, chaque unité mobile maintient une vision de la topologie ainsi que l'état des liens. Pour maintenir cette topologie à jour, chaque unité mobile diffuse l'état des liens de ses voisins. Une des défaillances de cette méthode est qu'elle surcharge le réseau par du trafic de contrôle et utilise plus d'espace pour le stockage.

Plusieurs protocoles utilisant cette méthode existent dans la littérature. Par simplification, on se propose de les citer et de ne décrire que deux de ceux qui font intervenir la notion de grappe introduite au premier chapitre. On dénombre les protocoles comme Wireless Routing Protocol (WRP) (Murthy et al., 1995), Destination

Sequence Distance Vector (DSDV) (Charles et al., 1994), Least Routing Protocol (LRP), Hierarchical State Routing (HSR) et le Clusterhead Gateway Switch Routing (CGSR) pour la première méthode. Les protocoles Global State Routing (GSR) (Tsu et Gerla, 1998), Fishe State Routing (FSR), Adaptive Link state protocol (ALP) (Garcia-Luna-Aceves et Spohn, 1998), Source Tree Adaptive Routing (STAR) (Garcia-Luna-Aceves et Spohn, 1999), Optimized Link State Routing (OLSR), utilisent la méthode *link state*. Dans les deux sous-sections qui suivent, nous présentons les fonctionnalités des protocoles HSR et CGSR.

2.2.1.1 Le protocole de routage HSR

Le protocole HSR (Routage à État Hiérarchique) a été proposé dans le but de résoudre les problèmes de mobilité et de localisation que rencontre le routage dans les réseaux sans fil. Il combine les notions de grappes dynamiques et de niveaux hiérarchiques avec une gestion efficace de la localisation. Dans ce protocole, le réseau est subdivisé en un ensemble de groupes au sein desquels un représentant est élu. L'ensemble des représentants forme à leur tour des groupes de niveau hiérarchique supérieur et le processus continue jusqu'à un groupe de deux unités mobiles. Cette partition du réseau permet d'avoir :

- une utilisation efficace du médium de communication ;
- une réduction du trafic de contrôle.

L'un des avantages du HSR est qu'il utilise une *adresse hiérarchique*. Cette dernière permet à une unité mobile de mettre à jour dynamiquement et localement son identificateur hiérarchique (HID) dès qu'elle reçoit des données d'une autre unité de niveau supérieur.

Le protocole HSR fait deux types de partitions, une *partition physique* et une *partition logique*. La partition logique joue un rôle important dans la localisation. Ainsi, une adresse logique comprenant le sous-réseau et un nom d'hôte est associée à chaque élément du réseau. Chaque sous-réseau correspond à un groupe particulier d'utilisateurs qui possèdent un serveur de localisation dit LMS (Location Management Server).

Dans l'exemple de la Figure 2.1, le réseau est subdivisé en trois groupes, G0-1, G0-2, G0-3 qui forment le niveau 0 de la hiérarchie. Les représentants des groupes des niveau 0 sont encore groupés pour former le niveau 1 et ainsi de suite. Dans la décomposition en groupes, on peut avoir un représentant de groupe nommé tête du groupe, une unité de liaison qui relie deux groupes et une unité interne qui n'a aucun rôle spécial. Une des méthodes utilisée pour associer des adresses hiérarchiques aux unités mobiles consiste à prendre les numéros des groupes dans le chemin reliant la racine et l'unité mobile en question. Par exemple, l'adresse HID (6) = <1,1,6> veut dire que le second 1 est le représentant de G0 tandis que le premier 1 est le représentant de G1.

2.2.1.2 Le protocole de routage CGSR

Dans cette section, nous décrivons brièvement le protocole CGSR (Clusterhead switch Routing). Ce protocole décompose l'ensemble des unités mobiles du réseau en grappes et choisit dans chaque grappe une tête de grappe. Les autres unités mobiles appartenant à la zone de portée de transmission de cette dernière deviennent des membres de cette grappe. Cette manière d'organiser n'est pas adaptée au réseau à topologie dynamique.

Le protocole CGSR utilise, en plus, un algorithme appelé LCC (Least Cluster Change) dans lequel un changement de tête de grappe a lieu si :

- il y a une fusion de deux grappes ;
- Une unité mobile sort complètement de la portée de transmission de toutes les têtes de grappes du réseau.

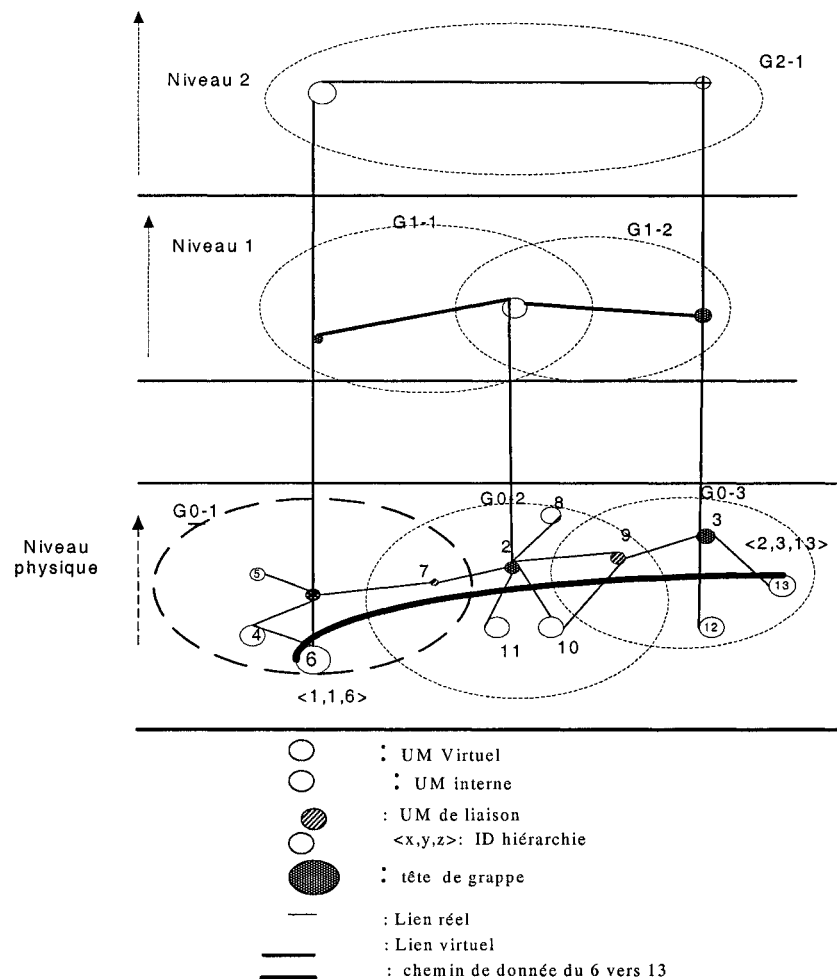


Figure 2.1 Partitionnement du réseau en groupes

Si une source S veut communiquer avec une destination D , elle transmet les paquets de données à la tête de grappe. Si S et D n'appartiennent pas à la même grappe, la tête de grappe envoie les paquets à la passerelle qui, à son tour, le transmet à la tête de grappe suivante. Le processus se répète jusqu'à ce qu'on atteigne la tête de grappe qui héberge la destination. Cette dernière le transmet à son tour à D .

Lors de la réception d'un paquet, la passerelle trouve la tête de grappe la plus proche sur le chemin vers la destination en utilisant la table des membres de la grappe et celle de routage. Cette manière de procéder

- assure un procédé déterministe et efficace pour l'acheminement des informations,
- peut cependant amener à un choix de route non optimal.

En effet, à la Figure 2.2, la route (9,2,10,3,12,4,15,5,18) entre la source 9 et la destination 18 est moins bonne que la route (9,2,10,3,12,4,15,18). En utilisant sa table de routage, l'unité mobile 15 trouve que l'élément suivant sur la route vers 18 est 18 elle-même. Par contre, si elle consulte sa table des membres de grappe, elle trouve que la tête de grappe hébergeant 18 est 5. Elle transmet les paquets vers 5 au lieu de les envoyer directement vers la destination.

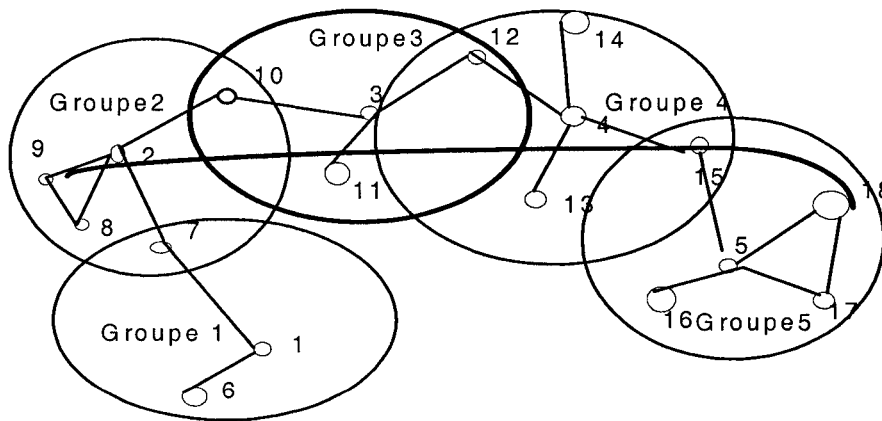


Figure 2.2 Illustration du protocole CGSR

2.2.2 Les protocoles de routage réactifs

Les protocoles de routage réactifs ne maintiennent pas des routes pour chaque paire source-destination. Ils les établissent à la demande. Ainsi, quand une source veut entrer en communication avec une destination quelconque, elle diffuse une requête de construction de route. Dès que la destination reçoit la requête, elle lance un algorithme de sélection de route pour déterminer la meilleure. Par la suite, la destination envoie une réponse à la source pour lui indiquer la route choisie. À la réception de la réponse, la source peut initier la communication.

Dans la littérature, nous retrouvons les protocoles comme Dynamic Source Routing (DSR), Temporary Ordered Routing Algorithm (TORA) (Park et Corson,

1999), Ad hoc On Demand Distance Vector (AODV), Associativity-Based Routing (ABR) (Toh, 1997), Relative Distance Micro-discovery Ad Hoc Routing (RDMAR) (Aggelou et Tafazolli, 1999) et le Cluster Based Routing Protocol (CBRP).

2.2.2.1 Le protocole de routage DSR

Le protocole DSR est basé sur l'utilisation de la technique "*routage source*" (Johnson et al., 1996). Dans cette technique, la source détermine la séquence complète d'unités mobiles à travers lesquelles les paquets de données seront envoyés. Elle inclut par la suite cette route dans l'entête du paquet de données. Dans le paquet de construction de route, la source indique les adresses des unités mobiles qui sont sur la route vers la destination. DSR comprend deux mécanismes de base, à savoir la *découverte de routes* (route discovery) et la *maintenance de routes* (route maintenance). La *découverte de routes* permet à la source de découvrir dynamiquement un chemin vers une destination. Dans ce mécanisme (Figure 2.3), la source diffuse un paquet « *requête de route* » identifiant la destination. Si l'opération de découverte est réussie, la source reçoit un paquet « *réponse de route* » qui liste la séquence des unités mobiles à travers lesquelles la destination peut être atteinte. La *requête de route* contient :

- l'adresse de la source,
- un champ *enregistrement de la route*,
- un identificateur unique de la requête.

À la réception d'une requête de route, l'unité mobile :

- rejette le paquet si elle l'avait déjà reçu, ce qui permet d'éviter les boucles de routage ;
- renvoie, dans un paquet « *réponse de route* », la route enregistrée à la source si c'est la destination, inversant pour cela le chemin spécifié dans l'*enregistrement de route*, comme illustré à la Figure 2.4 ;
- rajoute son adresse dans l'enregistrement de route du paquet reçu dans d'autres cas.

Dans la procédure de *maintenance de routes*, quand une unité mobile détecte une défaillance quelconque, elle envoie à la source un message « *erreur de route* ». Ce message contient l'adresse de l'unité mobile qui a détecté l'erreur et celle de l'unité mobile qui suit dans le chemin. Lors de la réception du paquet *erreur de route*, la source supprime du chemin sauvegardé l'unité mobile concernée par l'erreur. Elle initie par la suite une opération de découverte de route vers la destination.

Parmi les avantages du protocole DSR, on peut noter :

- que l'envoi des paquets de données par les unités mobiles de transit ne nécessite pas la maintenance des informations de mise à jour car ils contiennent toutes les décisions de routage ;
- une absence totale de boucle de routage, car le chemin source-destination fait partie des paquets de données envoyés.

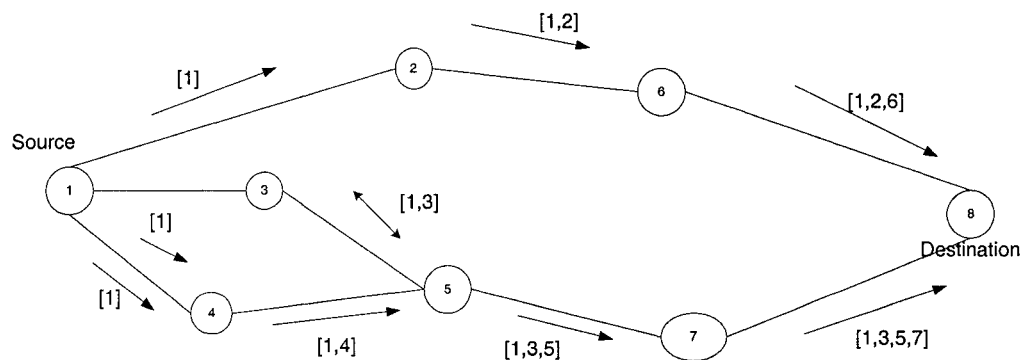


Figure 2.3 Construction de l'enregistrement de route

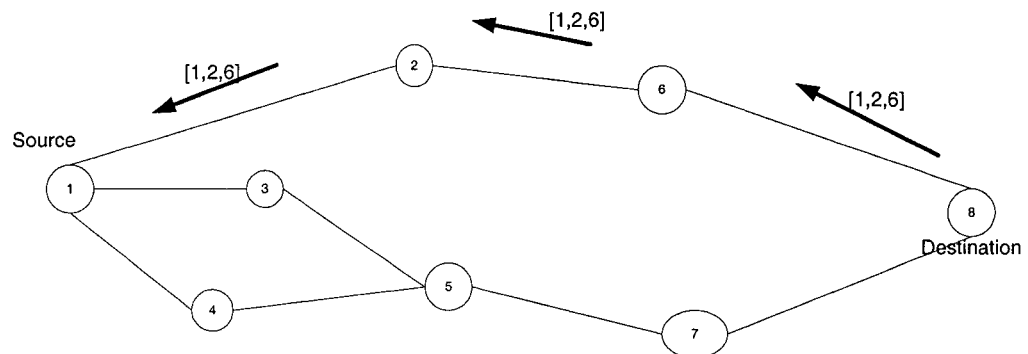


Figure 2.4 Renvoi de la route

2.2.2.2 Le protocole de routage AODV

Le protocole AODV (Charles et Elizabeth, 1999) détermine les routes à la demande, ce qui réduit de ce fait le nombre de messages diffusés dans le réseau. Dans cette section, nous décrivons brièvement les fonctionnalités de ce protocole.

Le protocole AODV se base :

- sur les mécanismes de *découverte de route* et de *maintenance de route* évoqués dans le protocole DSR ;
- sur le routage *nœud-par-nœud* ainsi que le principe des numéros de séquence.

Les numéros de séquence utilisés par le protocole AODV permettent :

- de maintenir la consistance des informations de routage. En effet, la mobilité fait que les routes changent fréquemment et par conséquent certaines routes maintenues ne sont plus valides ;
- d'utiliser les routes les plus nouvelles.

Pour construire une route entre une source (S) et une destination (D), le protocole AODV utilise une « *requête de route* » comme dans le protocole DSR. AODV maintient les routes d'une façon répartie, en gardant une table de routage, au niveau de chaque unité mobile de transit sur la route cherchée. Une entrée de la table de routage contient essentiellement l'adresse de la destination, l'adresse de l'unité mobile suivante, la

distance en nombre de sauts (hop), le numéro de séquence et le temps d'expiration de l'entrée de la table.

Quand une unité mobile de transit renvoie une requête à un voisin, elle garde l'identificateur de celle qui a envoyé la première copie de la requête reçue. Cela permet une construction aisée de la route vers la source, comme le montre la Figure 2.6. Cette route sera utilisée par le paquet *réponse de route*. Si une source veut connaître une route vers une destination, elle diffuse une *requête de route notée RREQ* (Route REQuest), comme le montre la Figure 2.5. La source attend par la suite le paquet *réponse de route* (RREP: Route REPlY) renvoyé par la destination (Figure 2.6). Si ce dernier n'est pas reçu durant une certaine période (appelée RREP_WAIT_TIME), la source peut rediffuser une nouvelle requête RREQ. Un message d'erreur est émis si la requête RREQ est rediffusée un certain nombre de fois sans recevoir de réponse.

Pour maintenir les routes, le protocole :

- transmet périodiquement des messages "*HELLO*" ;
- considère que le lien est défaillant si trois messages "*HELLO*" consécutifs sont perdus.

Quand un lien reliant une unité mobile à son voisin tombe en panne, cette unité mobile diffuse vers la source un paquet *UNSOLICITED RREP* avec un numéro de séquence initialisée à l'ancienne valeur du paquet RREP incrémentée de un et une valeur *infinie* de la distance. Dès que la source reçoit ce paquet, elle initie le processus de découverte de routes.

AODV présente quelques avantages:

- l'évaluation de performance montre des ressemblances du protocole (en terme d'optimisation) entre le chemin qu'il utilise et le chemin utilisé par les protocoles basés sur les algorithmes de plus court chemin (PCC) ;
- le protocole résout le problème du "*counting to infinity*", ce qui offre une convergence rapide quand la topologie change.

En revanche, le protocole n'assure pas l'utilisation du meilleur chemin entre la source et la destination.

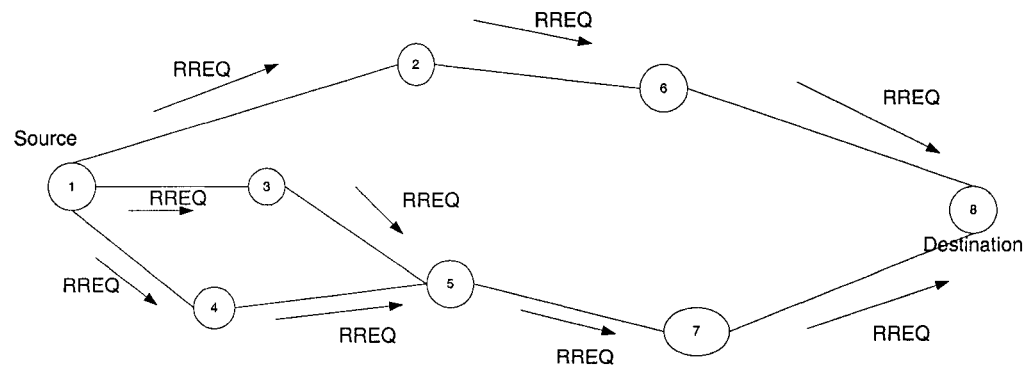


Figure 2.5 Diffusion du paquet Route Request (RREQ)

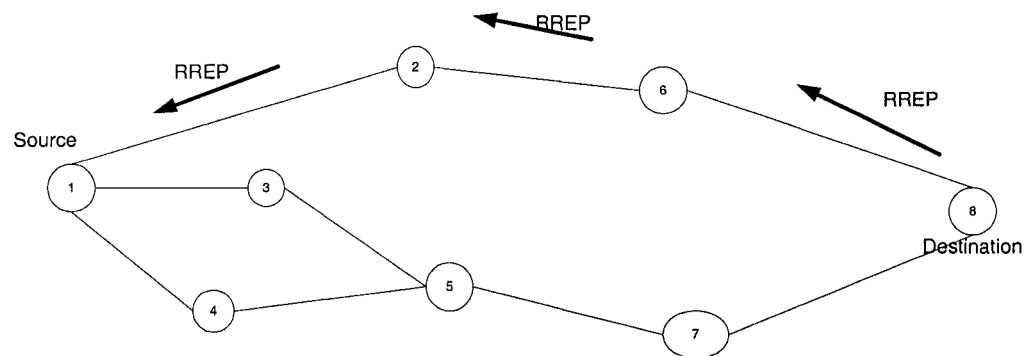


Figure 2.6 Renvoi du paquet de réponse (RREP)

2.2.2.3 Le protocole de routage CBRP

Dans le CBRP, les unités mobiles sont groupées en grappes (Jiang et al., 1999). Au départ, chaque unité mobile est dans l'état « indécidé ». Toute unité mobile qui a ce statut active un "timer" et diffuse un message "Hello". Lorsqu'une tête de grappe reçoit le message, elle envoie immédiatement une réponse à la source. Dès que la source reçoit la réponse, elle change de statut et devient "*membre*". Si après un délai de garde (un time-out) la source n'a toujours pas de réponse, elle se considère elle-même comme une tête de grappe.

Afin de sauvegarder la répartition des unités mobiles dans les grappes, chaque élément maintient une *table des voisins*. À la Figure 2.7, on illustre la subdivision du réseau par le protocole CBRP. À chaque voisin est associée une entrée qui indique son

statut (membre ou tête de grappe). La tête de grappe maintient les informations des membres qui appartiennent à sa grappe. Elle possède aussi une *table des groupes adjacents*. Une entrée dans cette table est associée à un groupe voisin et contient :

- l'identificateur du groupe,
- l'identificateur de l'élément de liaison à travers lequel le groupe peut être atteint.

Pour déterminer une route, la source diffuse une requête de route uniquement aux têtes de grappes voisines. Une tête de grappe qui reçoit une requête de demande de route, vérifie d'abord si la destination n'est pas dans sa table des membres de la grappe. Si oui, elle envoie directement la requête à la destination. Sinon, elle diffuse la requête aux têtes des grappes voisines.

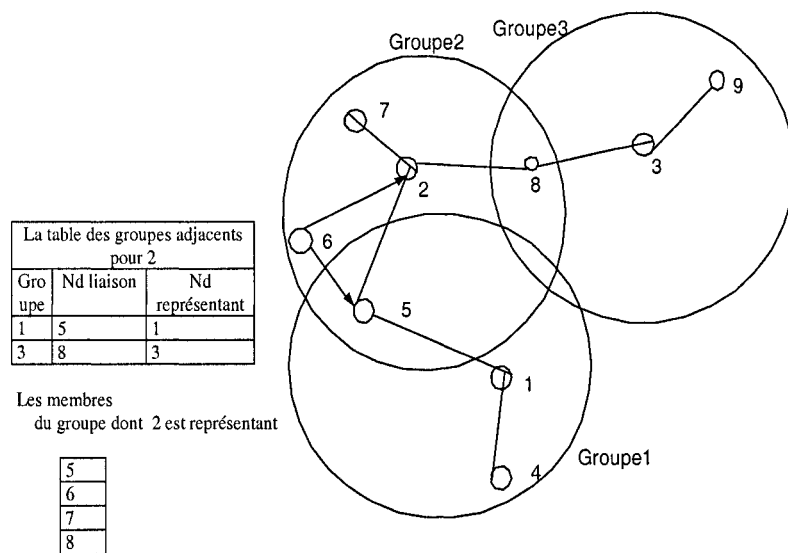


Figure 2.7 Organisation du réseau dans le CBRP

L'adresse des têtes de grappe est incluse dans la requête de route. Quand la destination reçoit la requête, elle renvoie à la source la route enregistrée. Si après un délai de garde la source ne reçoit pas de réponse, elle émet de nouveau la requête.

Lors de l'acheminement des données, si une unité mobile détecte qu'un lien est défaillant, elle fait retourner un message d'erreur à la source. Elle applique ensuite *un mécanisme de réparation locale*. Dans ce mécanisme, si une unité mobile trouve que

l'unité suivante ne peut pas être atteinte, elle essaie de vérifier si cette unité ou la suivante peut être atteinte en passant par une voisine. Si oui, les données sont envoyées en utilisant la route réparée.

2.2.3 Limitations des protocoles de routage dans les réseaux ad hoc

Les protocoles de routage proactifs et réactifs utilisent des messages de signalisation qui sont soit des échanges de tables de routage soit des requêtes de construction, de maintenance et de mise à jour des routes entre une origine et une destination. Les messages de signalisation de ces protocoles sont souvent transmis par diffusion (broadcast). Cela génère un trafic de contrôle excessif et consomme ainsi une bande passante déjà très réduite.

Malgré le fait que la route soit déterminée à l'avance dans les protocoles proactifs, le fait qu'ils échangent continuellement les informations de routage gaspille les ressources réseau. Ainsi, si le réseau est dynamique, les approches basées sur ce type de routage ne sont pas efficaces.

Le mécanisme de détermination de routes dans les protocoles réactifs introduit un délai supplémentaire et un trafic de contrôle excessif. En outre, il introduit un trafic volumineux par les requêtes de route. Il en découle une très mauvaise qualité de route par rapport au routage proactif. Ainsi, les protocoles purement réactifs ne conviennent pas aux applications temps réel.

Un compromis entre ces deux catégories est un protocole hybride qui, d'un côté, détermine la route sur demande et de l'autre impose un coût de recherche limité. Le protocole Zone Routing Protocol (ZRP) (Haas et Pearlman, 1997) a été élaboré sur la base de cette hypothèse.

Avec l'apparition du système GPS (Global Positionning System) (Capkun et al., 2002), des protocoles utilisant des informations de localisation ont été développés, en l'occurrence LAR (Young-Bae et Vaidya, 1998; Young-Bae et Vaidya, 2000) et DREAM (Basagni et al., 1998). Ces protocoles ont permis d'augmenter les performances quant à l'utilisation de la bande passante. En effet, le protocole LAR se

base sur les informations de localisation pour limiter la région de propagation des requêtes de construction de route. De son côté, le protocole DREAM fait des mises à jour périodiques des tables de localisation et diffuse les données dans la direction générale de la destination. Ces tables de localisation contiennent les coordonnées de chaque unité mobile ainsi que les vitesses et les directions de déplacement. La mobilité, l'optimisation de la bande passante et surtout les données contenues dans ces tables de localisation ont inspiré l'utilisation d'une méthode de construction et de maintenance des routes par la méthode de prédiction de la mobilité. Dans la section qui suit, nous décrivons brièvement cette dernière méthode et nous montrons également ses limites.

2.2.4 Routage par la méthode de prédiction de la mobilité

La méthode de prédiction de la mobilité (Lee et al., 1999) est utilisée pour prédire le temps d'expiration d'un lien (LET : Link Expiration Time) entre deux éléments se trouvant dans une même zone de portée de transmission. On déduit ainsi le temps d'expiration d'une route (RET : Route Expiration Time) entre une origine et une destination. Le RET est le minimum entre tous les LET des liens qui composent la route. À partir du RET et du temps de transmission du dernier paquet noté « T_d », on évalue le temps critique noté « T_c » pour une route ($T_c = RET - T_d$). La source devra donc reconstruire une route de secours (back-up ou hand-off) avant l'expiration de T_c , ce qui réduit la surcharge (overhead) de contrôle.

La méthode de prédiction de la mobilité permet d'une part de rendre un protocole de routage plus robuste à la mobilité et d'autre part une utilisation efficace de la bande passante. Les paramètres de la mobilité sont souvent analysés pour prédire l'état futur de la topologie du réseau et reconstruire les routes de secours de façon proactive. De cette façon, on élimine la transmission des paquets de contrôle qui consomment la bande passante et par conséquent peuvent causer la congestion.

2.2.4.1 Temps d'expiration d'un lien et d'une route

Après une brève introduction sur la méthode de prédiction de la mobilité, nous présentons dans cette section la méthode de calcul du temps d'expiration d'un lien (LET) et du temps d'expiration d'une route (RET) puis nous l'illustrons par un exemple. Supposons :

- que le système GPS fournit les informations de localisation et les caractéristiques du mouvement de deux éléments i et j , en l'occurrence les coordonnées de i (x_i, y_i) et celles de j (x_j, y_j), leurs vitesses v_i et v_j , leurs directions de déplacement et la portée de transmission r ;
- que la propagation se fasse en espace libre et que les horloges de toutes les unités mobiles soient synchronisées par le protocole NTP (Network Time Protocol).

Nous déterminons la durée de disponibilité d'un lien de communication entre i et j par la formule (2.1). Dans cette formule, si $v_i = v_j$, et $\theta_i = \theta_j$, alors $a = c = 0$ et la durée de disponibilité du lien est infinie.

$$LET = \frac{-(a*b + c*d) + \sqrt{(a^2 + c^2)r^2 - (a*d - b*c)^2}}{a^2 + c^2} \quad (2.1)$$

où

$$a = v_i \cos \theta_i - v_j \cos \theta_j,$$

$$b = x_i - x_j,$$

$$c = v_i \sin \theta_i - v_j \sin \theta_j,$$

$$d = y_i - y_j,$$

$$RET = \frac{1}{n} \sum_{i=0, i \in N(v)}^n LET(\text{tête de grappe et } i)$$

La méthode de prédiction de la mobilité est utilisée pour les protocoles tant *unicast* que *multicast*. Dans l'exemple de la Figure 2.8, il y a deux routes entre la source A et la destination F . La première route est $(ABCEF)$ dont les valeurs des LET sont 4, 4, 3, et 6, ce qui donne un RET de 3. La seconde est $(ABDEF)$ avec des valeurs des LET

de 4,5, 4 et 6, d'où un RET de 4. *F* renvoie donc à *A* un flot de sélection de route (*Flow-Setup*) contenant la route *ABDEF*.

En analysant cette méthode, nous constatons que, malgré le fait que la route choisie soit stable pendant une certaine période, elle peut ne pas répondre au critère de délai exigé, surtout par les applications temps réel. En outre, elle n'a pas inclus la contrainte sur la capacité d'un lien ou du tampon au niveau des unités mobiles de transit.

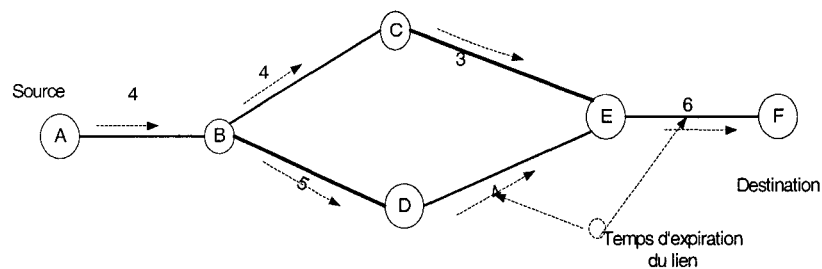


Figure 2.8 Flux de requête (Flow-REQ)

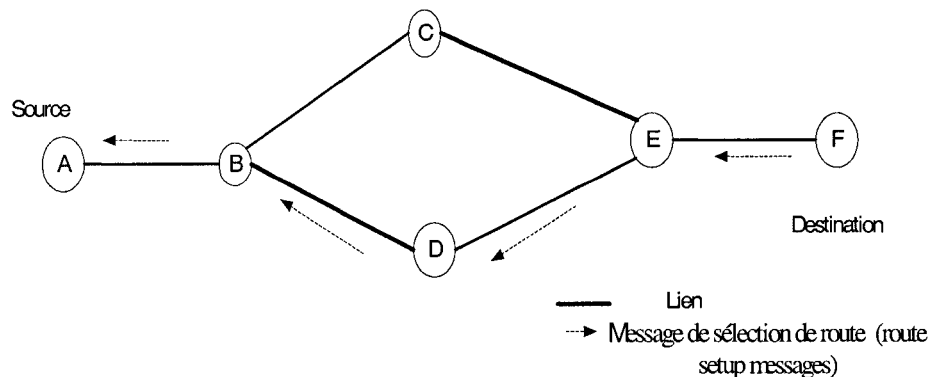


Figure 2.9 Flux de sélection de route (Flow-SETUP)

2.2.5 Routage multicast dans les réseaux ad-hoc

Dans les réseaux mobiles ad hoc, on distingue deux types de communication, à savoir les communications *unicast* (i.e un à un) et les communications *multicast* (i.e un à plusieurs ou plusieurs à plusieurs). Dans cette section, nous faisons un bref aperçu sur certains protocoles de routage multicast déjà développés pour les réseaux mobiles ad

hoc. Ces protocoles sont basés sur deux approches principales : la construction des arbres de routage et la construction des graphes de routage.

Approche 1 : la construction des arbres de routage

Un arbre de routage est construit pour un groupe de routeurs, comme le montre la Figure 2.10. Les protocoles de cette catégorie ont une simplicité et une efficacité dues au fait qu'ils forcent une route entre toute paire de routeurs. Mais, si plusieurs sources doivent transmettre vers un même ensemble de destinations, deux scénarios existent :

- soit l'arbre est partagé,
- soit chaque élément de la route maintient une entrée pour chaque source.

Chacun des deux scénarios a un inconvénient. En effet, le partage peut induire un délai supplémentaire tandis que le deuxième scénario limite la croissance du réseau. Les protocoles de cette catégorie sont utilisés dans la majorité des réseaux informatiques fixes et dans l'Internet.

Par contre, la maintenance de ces routes dans les réseaux à mobilité élevée serait coûteuse. En effet, elle exige des mises à jour fréquentes qui induisent un trafic de contrôle excessif. En outre, pendant les périodes de reconstruction de ces arbres, un routeur peut arrêter de transmettre le paquet et imposer de ce fait des délais supplémentaires. On retrouve ici les protocoles comme Ad hoc *Multicast* Routing (AMRoute) (Bommaiah et al., 1998) et Ad Hoc Multicast Routing Protocol using Increasing id-numberS (AMRIS) (Wu et Tay, 1999).

Approche 2 : la construction des graphes de routage

Une maille offre plus de connectivité, c'est-à-dire des chemins multiples entre une source et une destination, comme le montre la Figure 2.11. On retrouve dans cette catégorie les protocoles comme On-Demand Multicast Routing Protocol (ODMRP) (Lee et al., 1999), Core-Assisted Mesh Protocol (CAMP) (Garcia-Luna-Aceves et al., 1999) et le Forwarding Group Multicast Protocol (FGMP) (Chiang et al., 1998).

Dans la suite de cette section, nous présentons les fonctionnalités du protocole ODMRP.

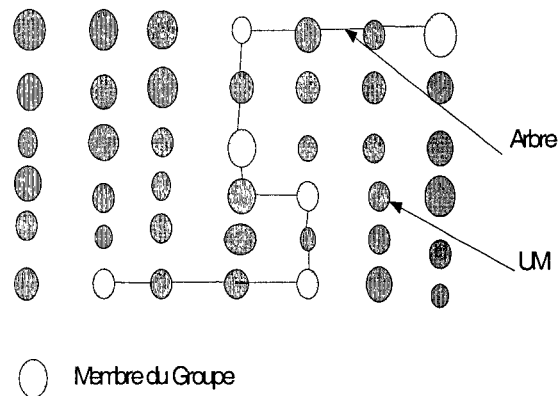


Figure 2.10 Illustration des protocoles basés sur l'approche d'arbre

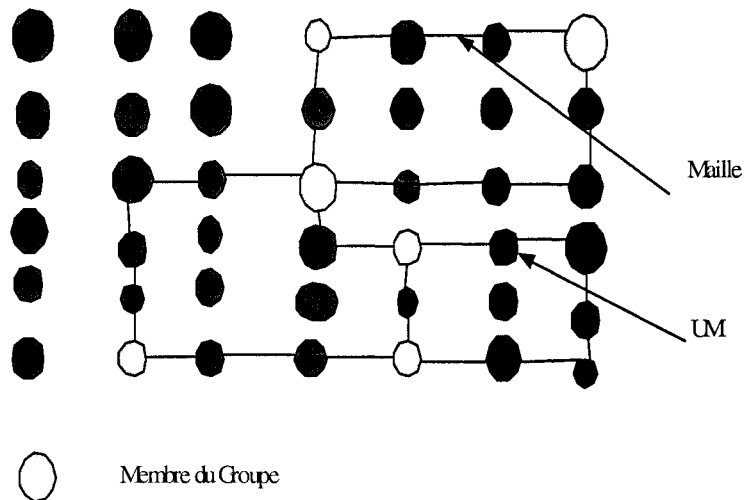


Figure 2.11 Illustration des protocoles basés sur l'approche de maille

2.2.6 Description de ODMRP

ODMRP (On-Demand Multicast Routing Protocol) est un protocole réactif basé sur l'approche de maille *multicast* comme le montre la Figure 2.12. Avec une topologie

maillée, les paquets *multicast* peuvent toujours arriver à destination malgré la mobilité des éléments et le changement de la topologie du réseau. ODMRP crée un ensemble appelé «*forwarding group*» ou *groupe de multi-diffusion*, comme le montre la Figure 2.13. Cet ensemble diffuse les paquets *multicast* sur les plus courts chemins entre toute paire origine–destination. Si une source a des données à envoyer, elle diffuse périodiquement une requête «JOIN REQUEST» pour déterminer et maintenir les routes *multicast*. Toute autre unité mobile qui reçoit une requête non dupliquée stocke dans sa cache l'identificateur (ID) de son prédécesseur et le numéro de séquence de la requête avant de la rediffuser par la suite. Seul un nœud du groupe *multicast* qui reçoit cette requête répond en diffusant une table «JOIN TABLE».

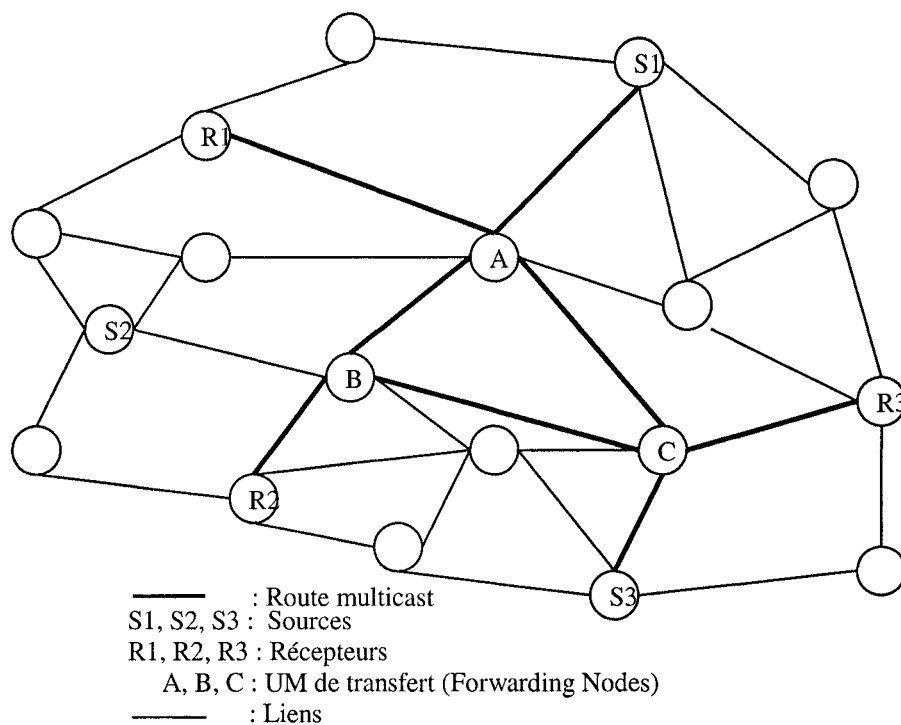


Figure 2.12 Illustration d'une maille

Quand la requête arrive à destination, cette dernière crée un « JOIN REPLY » qu'elle diffuse à ses voisins. La voisine qui reçoit le « JOIN REPLY » vérifie si, parmi les entrées, il y en a une dont l'élément suivant correspond à son propre identificateur

(ID). Si oui, il réalise qu'il est sur le chemin qui mène à la source. Elle initialise une étiquette du *forwarding group* (FG-Flag) et diffuse son propre « JOIN REPLY ». Le protocole ODMRP utilise l'approche «soft state», i.e. le rafraîchissement des liens et la présence ou l'absence d'une unité mobile déterminée à l'aide d'un délai de garde (time-out).

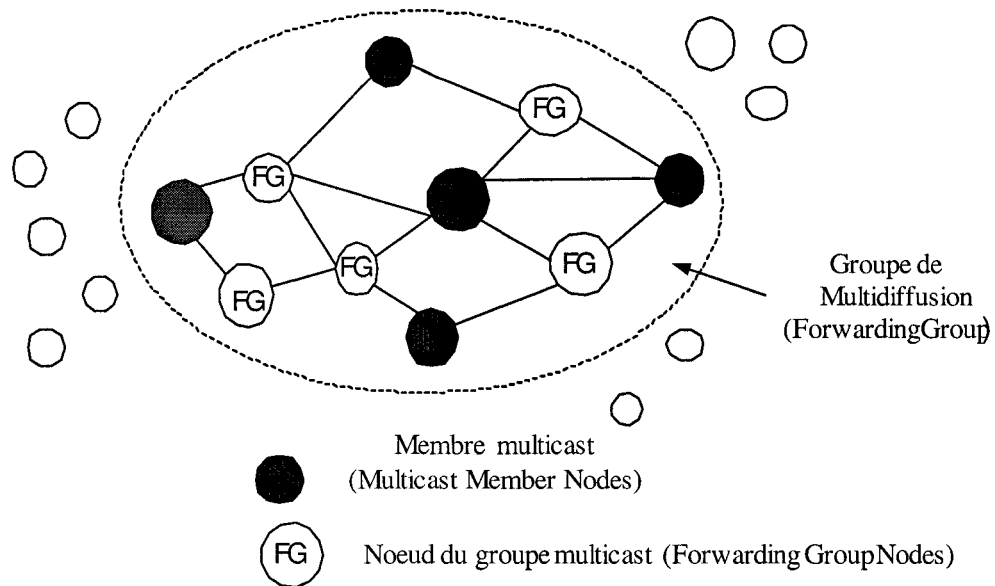


Figure 2.13 Concept de groupe de multi-diffusion

Si une unité mobile arrête d'envoyer les paquets « JOIN QUERY » et « JOIN REPLY » pendant un temps qui excède le délai de garde (time-out), elle est considérée comme étant en dehors du groupe. Le protocole ODMRP n'utilise pas des paquets de contrôle pour entrer ou sortir du groupe et peut être utilisé par des opérations *unicast* et *multicast*. Pour le protocole ODMRP, un champ supplémentaire est ajouté au paquet "JOIN DATA" et "JOIN TABLE". Si une source envoie le paquet "JOIN DATA", elle ajoute les informations de localisation et de vitesse (ses coordonnées x, y, sa vitesse et sa direction de déplacement). Elle ajoute également un champ appelé MIN-LET qui contient le RET. Ce champ est initialisé à la valeur MAX-LET-VALUE si la source n'a pas de prédécesseur. Le voisin suivant, qui reçoit le paquet JOIN DATA, prédit le temps d'expiration du lien entre lui et son prédécesseur en utilisant la formule (2.1). Le

minimum entre cette valeur et le MIN-LET indiqué dans le paquet “JOIN DATA” est inclus dans ce dernier. L’un des défauts de cette amélioration est que, aussitôt qu’un des liens de la route est coupé, toute la route devient invalide.

Le voisin remplace par ses propres valeurs les anciennes valeurs de localisation et les informations de mobilité écrites par son prédécesseur. Quand un membre *multicast* reçoit le paquet “JOIN DATA”, il calcule le LET du dernier lien sur le chemin. Le minimum entre le LET du dernier lien et le MIN-LET spécifié dans le “JOIN DATA” est le RET. Il est mis dans le paquet “ JOIN TABLE ” avant que ce dernier ne soit diffusé. La destination choisit la route la plus stable (le plus grand RET) au lieu de la route la plus courte (en termes de délai).

2.3 Algorithmes de subdivision en grappes

La majorité des algorithmes de routage utilise la diffusion (broadcast) dans la procédure de détermination d’une route entre une source et une destination. C’est un mécanisme qui convient à ce type de réseaux d’autant plus qu’il n’exige aucune connaissance préalable de la topologie du réseau. Cependant, c’est un mécanisme qui génère un nombre important de messages redondants, consommant ainsi les ressources réseaux déjà limitées, en l’occurrence la bande passante et l’énergie. La pénalité augmente avec la taille et la densité du réseau. Ainsi, des méthodes de réduction des participants au processus de rediffusion ont été proposées. Elles sont basées sur deux approches :

- La première construit soit un arbre enraciné à la source avec un nombre maximal de nœuds sans fil ou une maille. Cette méthode exclut les nœuds sans fil des participants dans la rediffusion.
- La seconde approche est basée sur une structure en grappes à deux sauts.

Dans les sections suivantes, nous parlons de la notion de mise en grappe et de quelques algorithmes d’élection des têtes de grappes déjà présentés dans la littérature.

2.3.1 La mise en grappe

La mise en grappe est une manière de grouper les UM dans un ensemble facile à gérer. Il permet d'améliorer la gestion du réseau, l'efficacité du canal de transmission et l'économie d'énergie. Plus encore, la mise en grappe est indispensable dans le routage hiérarchique.

Plusieurs mécanismes de mise en grappe ont été proposés dans la littérature. Tous supposent une connaissance préalable des voisins. La majorité a été testée par simulation car un autre environnement exige la connaissance de la localisation de toutes les UM et des informations sur les voisins, ce qui est difficile. Ces informations varient avec le temps. La collecte de toutes ces informations se fait par des échanges périodiques de messages de signalisation et, pendant cette phase, les UM sont supposées fixes.

Dans (Kwon et Gerla, 2002), on a introduit la mise en grappe passive. C'est une approche innovatrice qui, au lieu d'utiliser un protocole spécifique de signalisation, applique dans un champ réservé de l'entête de tout paquet transmis, l'information sur le statut de l'unité mobile (initial, tête de grappe, passerelle ou nœud ordinaire). Ainsi, une UM collecte les informations à partir de l'adresse MAC de l'émetteur amenée par les paquets entrants et peut construire les grappes sans collecter la liste des voisins.

Dans (Vaidya et al., 1997), un algorithme qui organise de façon dynamique la topologie du réseau en k -grappes (k -clusters) est proposé. Dans ces grappes, les unités mobiles communiquent via k sauts. L'algorithme considère $k = 1$ et essaie de trouver des cliques dans la topologie physique. Ensuite, il exécute une heuristique appelée « first-fit » pour trouver les cliques les plus larges.

Le défaut de cet algorithme est qu'il ne forme pas de grappes optimales. En effet, quand un changement survient dans le réseau, l'algorithme nécessite toujours trois étapes : il trouve d'abord un ensemble de grappes réalisables, détermine ensuite les cliques les plus larges et les plus essentielles à la maintenance de la connectivité de tout le réseau avant d'éliminer ceux qui sont superflus.

Dans (Lin et Gerla, 1997), le schéma proposé utilise les grappes non pas pour minimiser la réaction du réseau aux changements topologiques mais pour promouvoir le contrôle d'accès à la bande passante et l'alternance des unités mobiles dans chaque grappe, afin de fournir une qualité de service. L'algorithme se base sur un numéro d'identification (ID) des unités mobiles pour construire les grappes.

Dans (Haas et Pearlman, 1997), un protocole ZRP (Zone Routing Protocol) est proposé. Ce protocole est une stratégie hybride dont l'objectif est de maintenir le routage proactif dans une zone et un routage réactif pour les communications inter-zones. Le défaut de ce protocole est qu'il ne reflète pas une mesure quantitative de la stabilité et les zones s'interfèrent de manière arbitraire.

Dans (McDonald et Znati, 1999), une stratégie dynamique pour organiser la topologie du réseau est proposée. Son objectif est de balancer les avantages du routage proactif et du routage réactif en groupant les unités mobiles dans des grappes sur la base de leur mobilité. Cette stratégie se base sur un algorithme réparti asynchrone qui maintient les grappes satisfaisant un critère (α, t) . Ce critère définit une probabilité α de disponibilité d'un chemin pendant un intervalle de temps t . Dans les réseaux à faible mobilité, cette stratégie est plus proactive tandis qu'elle est plus réactive dans le cas contraire. D'autres recherches se sont orientées dans la détermination d'un ensemble dominant d'unités mobiles.

2.3.2 Quelques algorithmes d'élection des têtes de grappes

Les algorithmes de formation de grappes permettent d'élire des têtes de grappes qui sont responsables de la formation des grappes. Parmi les algorithmes déjà proposés dans cette logique, on retrouve dans (Gerla et Tsai, 1995; Chatterjee et al., 2002) le "lowest-ID algorithm", le "highest-ID algorithm", le "Node-Weight heuristic" ainsi que le "WCA" (A Weighted Clustering Algorithm for Mobile Ad hoc Networks).

L'algorithme "lowest -ID algorithm" assigne d'abord à chaque unité mobile un identificateur (ID). Il choisit ensuite comme tête de grappe l'unité mobile qui a le plus petit ID, ce qui veut dire que tous ses voisins ont des ID plus grands. Le défaut de cet

algorithme est que les identificateurs (ID) sont attribués aux UM de façon arbitraire, sans considérer la possibilité d'une UM à devenir une tête de grappe. Comme les ID ne changent pas souvent, les têtes de grappes ont tendance à garder cette responsabilité pendant plus de temps que nécessaire.

Dans le “highest-ID algorithm”, l'unité mobile ayant le maximum de voisins (c'est-à-dire plus de connectivité) est élue comme tête de grappe. Dans cet algorithme, une unité mobile peut avoir plus de voisins qu'elle ne peut supporter à cause des limitations des ressources, même s'ils sont ses voisins directs et qu'ils se trouvent dans une zone délimitée par sa portée de transmission. D'un autre côté, un nombre élevé de têtes de grappes implique un coût excessif de calcul du système. Il introduit également une latence.

Le troisième algorithme (“Node-Weight heuristic”) assigne à chaque unité mobile un poids en se basant sur ses capacités à agir comme tête de grappe. Le principal défaut de cet algorithme est qu'il ne prend pas en compte le temps d'exécution de l'algorithme d'élection pendant lequel une unité mobile attend les réponses de ces voisins pour prendre la décision d'être une tête de grappe ou un nœud ordinaire.

Aucun de ces trois algorithmes ne fournit une élection optimale des têtes de grappes car ils prennent en compte un sous-ensemble de paramètres, ce qui peut imposer des contraintes au système. Chacun des trois est adapté à une application particulière au lieu d'un réseau sans fil arbitraire.

À partir des limitations de ces trois algorithmes, Chatterjee et al. (2002) ont développé un algorithme nommé WCA (A Weighted Clustering Algorithm for Mobile Ad hoc Networks). Cet algorithme prévoit un partage de charge et un seuil prédéfini d'unités mobiles qu'une tête de grappe peut supporter de façon idéale. Cela assure qu'aucune tête de grappe n'est surchargée à un instant donné. Un facteur est également défini pour mesurer le degré de partage de charges entre grappes. On définit également la connectivité comme la probabilité pour qu'une unité mobile soit atteignable à partir de n'importe quelle autre unité mobile.

Le WCA fournit de bons résultats par rapport aux trois autres en termes de nombre de ré-affiliations. Dans la section suivante, nous détaillons l'algorithme WCA.

2.3.3 Brève description de l'algorithme

L'algorithme WCA a été élaboré en vue de prendre en considération plusieurs paramètres systèmes dans le processus d'élection des têtes de grappes. En effet, il optimise le nombre de voisins qu'une tête de grappe peut supporter, la puissance de transmission, la mobilité ainsi que la puissance de la batterie de l'unité mobile.

Certaines applications peuvent nécessiter plusieurs têtes de grappes (donc plusieurs clusters). Cependant, un paquet envoyé d'une source à une destination aura un délai supplémentaire, une consommation excessive de puissance à cause d'un nombre élevé d'unités mobiles intermédiaires. On peut également minimiser le nombre de têtes de grappe tout en couvrant la population entière. Son avantage est une utilisation maximale de ressources. Cependant, il y a un nombre minimum de têtes de grappes qu'on doit avoir pour couvrir tout le réseau. Cette limite exige une distribution uniforme d'unités mobiles dont le nombre par grappe doit respecter les capacités des têtes de grappes. Dans le protocole WCA, la procédure d'élection est invoquée le plus rarement possible. Pour assurer un fonctionnement efficient du protocole de contrôle d'accès au médium de communication (MAC), un seuil d'unité mobile qu'une tête de grappe peut supporter est prédéfini.

L'utilisation de la batterie est efficiente si les UM se trouvent à l'intérieur de la zone délimitée par leur portée de transmission. À cause de ses responsabilités, la tête de grappe consomme plus d'énergie qu'un nœud ordinaire.

La mobilité, quant à elle, est un facteur important dans la procédure d'élection. En effet, une tête de grappe doit se déplacer le moins rapidement possible par rapport aux autres UM pour minimiser les réaffiliations.

La procédure d'élection d'une tête de grappe consiste en 8 étapes.

Étape 1 :

Pour chaque UM, il faut trouver tous ses voisins (i.e. toutes les UM qui communiquent directement avec elle). Ces derniers définissent son degré. Celui-ci est déterminé par la formule (2.2) où V désigne l'ensemble d'unités mobiles telles que, si v et v' sont deux éléments quelconques de cet ensemble, la distance entre elles doit être inférieure à la portée de transmission :

$$d_v = |N(v)| = \sum_{v' \in V, v' \neq v} \{dist(v, v') < txrange\} \quad (2.2)$$

Étape 2 :

Pour chaque UM et un seuil δ donné, on calcule la différence entre le degré déterminé à l'Étape 1 et le seuil δ en utilisant la formule (2.3) :

$$\Delta_v = |d_v - \delta| \quad (2.3)$$

Étape 3 :

Pour chaque UM, on calcule, par la formule (2.4), la somme de toutes les distances qui le sépare de ces voisins :

$$D_v = \sum_{v' \in N(v)} \{dist(v, v')\} \quad (2.4)$$

Étape 4 :

En utilisant la formule (2.5), on mesure la mobilité M_v de chaque UM :

$$M_v = \frac{1}{T} \sum_{t=1} \sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2} \quad (2.5)$$

Les différents termes de la racine sont les coordonnées de l'UM v aux instants t et $(t-1)$

Étape 5 :

On calcule le temps cumulatif pendant lequel une UM agit comme une tête de grappe; ce temps indique la puissance de la batterie consommée, étant donné que la tête de grappe en consomme plus qu'un *nœud ordinaire*, soit P_v .

Étape 6 :

On calcule le poids combiné des différents paramètres avec un facteur de poids correspondant à chaque paramètre en utilisant la formule (2.6):

$$W_v = w_1\Delta_v + w_2D_v + w_3M_v + w_4P_v \quad (2.6)$$

Étape 7 :

L'UM ayant la plus petite valeur de w_v est élue comme tête de grappe.

Étape 8 :

On répète les Étapes 2 à 7 pour les UM qui n'ont pas encore de statut (soit une tête de grappe ou assigné à une tête de grappe).

2.4 Architecture de base

Dans cette section, nous décrivons brièvement l'architecture de base qui a été utilisée pour l'étude des protocoles adaptés à la notion de grappe (clustering protocol). Nous prenons l'exemple du protocole ARC (Adaptive Routing using Clusters) (Elizabeth, 2002). Ce dernier sert d'interface entre la couche IP et un protocole de routage afin d'améliorer la consistance des routes dans les réseaux dynamiques. Dans cette architecture, deux types d'adressage peuvent être utilisés : l'*adressage de base* et l'*adressage hiérarchique*.

Dans l'*adressage de base*, les routes sont enregistrées saut par saut. À la Figure 2.14, on peut avoir entre la source S et la destination D la route S -CLH1-G1-CLH2- D . Une réparation de route intervient chaque fois qu'un lien est coupé.

Dans l'*adressage hiérarchique*, la seule route qui peut être enregistrée est S -CLH1-CLH2- D . Dans cette catégorie, n'importe quelle passerelle peut être utilisée pour router les paquets entre les deux têtes de grappe.

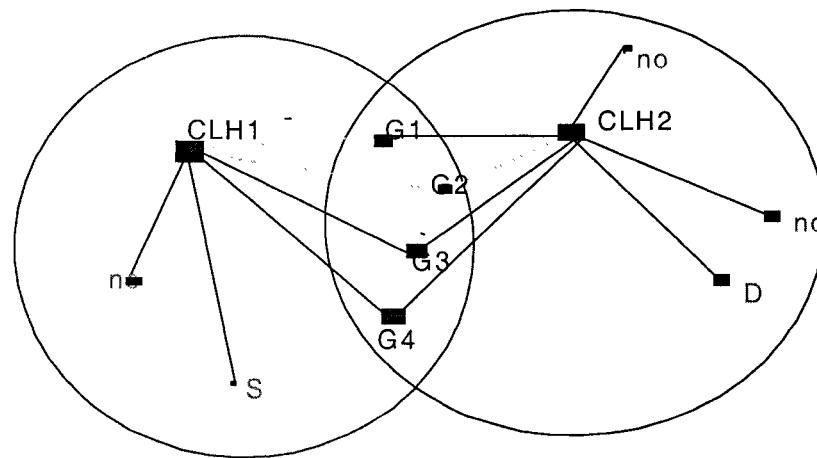


Figure 2.14 Adressage hiérarchique

2.4.1 Brève description du protocole ARC

Dans le protocole ARC, les unités mobiles sont groupées dans des grappes sur la base de leur proximité. Dans chaque grappe, toutes les unités mobiles sont en communication directe avec la tête de grappe. Un élément maintient soit une seule table soit deux tables en fonction de son statut. Dans ce qui suit, nous décrivons les différents cas.

- Si l'unité mobile est une tête de grappe, elle maintient deux tables : la première contient toutes les unités mobiles dont elle est responsable et la seconde contient toutes les têtes de grappe qui lui sont voisines ainsi que les passerelles qu'elle peut utiliser pour les atteindre.
- Les unités mobiles qui ne sont pas des têtes de grappe maintiennent seulement une seule table appelée table des têtes de grappe. Chaque unité mobile a autant d'entrées que de grappes auxquelles elle fait partie, ainsi que les grappes dont les têtes de grappe peuvent être atteintes à travers des passerelles réparties.

2.4.2 Étape d'initialisation

La topologie de chaque grappe est initialisée et maintenue à travers des messages de signalisation (*beacon*). Une unité mobile qui entre dans le réseau diffuse un message

« *hello* » pour annoncer sa présence et trouver des têtes de grappe qui lui sont voisines. Dans ce message, on retrouve l'adresse de la source (*Source_Addr*), le statut de la source (*Source_Status*), une liste de têtes de grappe avec lesquelles l'unité mobile peut directement communiquer et une liste de têtes de grappe avec lesquelles l'unité mobile peut seulement communiquer avec des passerelles réparties. Les réponses sont enregistrées dans la table des têtes de grappe.

À la fin de la période d'initialisation, une unité mobile qui n'a reçu qu'une réponse d'une seule tête de grappe devient un nœud ordinaire (*no*) et donc un membre de cette *grappe*. Si par contre elle reçoit plusieurs messages, elle devient une passerelle entre les différentes grappes. Cependant, si elle n'a reçu aucune réponse issue d'une tête de grappe, elle devient elle-même une tête de grappe.

2.4.3 Découverte de la topologie

Dans cette section, nous décrivons comment les unités mobiles déterminent la topologie du réseau ainsi que leur statut. Un élément peut être une tête de grappe, une passerelle ou un nœud ordinaire dans un réseau ad hoc. La topologie est déterminée à l'aide des messages de signalisation nommés « message hello ». Ces messages peuvent être échangés entre une tête de grappe et un nœud ordinaire, entre deux têtes de grappe et entre deux nœuds ordinaires.

Dans tous ces cas, dès qu'une unité mobile reçoit un message d'une autre unité mobile, elle vérifie d'abord si elle a une entrée correspondante dans sa table des unités mobiles. Si elle n'en a pas, elle en crée une. Dans ce qui suit, nous discutons des différents cas.

Une tête de grappe communique avec un nœud ordinaire

La tête de grappe vérifie alors si la liste des têtes de grappe avec lesquelles l'unité mobile peut communiquer concorde avec ce qui est reporté dans la liste des têtes de grappe du « message hello ». Cette liste indique l'ensemble d'unités mobiles pour lesquelles elle joue le rôle de passerelle. S'il y a une différence, la tête de grappe fait des

mis à jour appropriées et s'assure qu'il y a une entrée pour chaque tête de grappe listée dans le message hello. Si aucune liste de passerelles dans la table de voisins n'inclut l'unité mobile et si l'unité mobile ne liste pas cette tête de grappe dans le «message hello », l'unité mobile est effacée de la liste.

Une tête de grappe reçoit un « message hello » d'une autre tête de grappe

Cette tête de grappe met une étiquette sur cette entrée pour indiquer que le statut de cette unité mobile doit être analysé afin de déterminer quand il y aura fusion des deux grappes.

Un « nœud ordinaire » reçoit un «message hello » d'un autre « nœud ordinaire »

L'unité mobile s'assure que l'autre unité mobile est sa passerelle vers les têtes de grappe de la liste fournie par le «message hello». Elle vérifie de façon similaire que l'unité mobile n'est pas une passerelle pour une tête de grappe qui n'est pas dans la liste précédente. Si l'unité mobile ne peut atteindre la tête de grappe en utilisant des passerelles ni directement, elle supprime l'entrée relative à la tête de grappe de la table correspondante. Son statut reste « nœud ordinaire » au cas où elle ne peut toujours communiquer avec plus d'une tête de grappe.

CHAPITRE III

ARCHITECTURE ET ALGORITHMES DE ROUTAGE PROPOSÉS

D'une manière générale, lorsqu'un réseau mobile ad hoc est dense, on procède pour son étude à une mise en grappe et, dans chaque grappe, une tête de grappe est élue. Cette dernière devient responsable de la grappe. Certains paramètres évoqués au chapitre précédent ont fait l'objet de critères d'élection de cette tête de grappe. L'un des problèmes d'une telle approche est d'avoir traité isolément les sous-problèmes, alors qu'il y a une interaction certaine entre ces derniers. En effet, dans ces réseaux, on peut avoir des applications temps réel dont la contrainte principale est le délai ou les applications du moindre effort qui n'exigent que la fiabilité de la transmission. Comme contrainte, la bande passante est limitée, ce qui exige un protocole de routage nécessitant moins de messages de signalisation dans l'établissement et la maintenance d'une route entre une origine et une destination. En outre, le nombre de têtes de grappe dans le réseau a une influence sur les délais de transmission. Ce nombre varie en fonction de la portée de la transmission qui, elle-même, varie en fonction de l'interface de communication utilisée en l'occurrence les antennes omnidirectionnelles et directionnelles (Young-Bae et al., 2000). Dans ce chapitre, nous proposons une architecture et des algorithmes de routage pour garantir une qualité de service dans les réseaux mobiles ad hoc denses.

3.1 Principes de l'architecture proposée et hypothèses sous-jacentes

Même si, en principe, on affirme que dans les réseaux ad hoc, toutes les unités mobiles peuvent se comporter comme des routeurs, il demeure que celles-ci n'ont pas les mêmes potentialités à bien assurer la tâche de routage. En effet, il est très évident qu'un laptop (30 GB, 1.2 MHz, Pentium III) peut mieux assurer la fonction de routage qu'un simple PDA ou encore plus qu'un téléphone portable. D'un autre côté, les unités mobiles qui forment le réseau ad hoc peuvent être dotées du système de positionnement global GPS. La mobilité est souvent calculée par rapport au référentiel associé à ce

système. Cette mobilité a fait l'objet de critères de choix d'une tête de grappe (McDonald et Znati, 1999). Cependant, une unité mobile qui se déplace lentement dans ce référentiel peut s'écarter plus facilement de ses voisins si ces derniers sont très mobiles ou vice versa. D'où l'idée de considérer la mobilité relative. Dans cette section, nous exposons dans un premier temps l'idée de base qui sous-tend l'architecture que nous proposons et, dans un deuxième temps, les hypothèses sur lesquelles celle-ci repose.

3.1.1 Idée de base de l'architecture

Notre proposition s'inspire des observations faites dans le domaine militaire et celui de la médecine. Ces domaines sont parmi les principaux champs d'application des réseaux mobiles ad hoc. Dans le domaine militaire, on prend l'exemple d'une coalition de plusieurs armées devant effectuer une mission d'intervention des Nations Unies dans un lieu sans infrastructures de télécommunications comme le désert. Chaque armée est constituée de bataillons dans lesquels des commandants donnent des ordres. Il y a également dans chaque bataillon quelqu'un qui remplace le commandant en cas de problèmes, des agents de transmission et de simples soldats qui n'exécutent que les ordres. Pour le commandant, il est important de savoir à tout instant la localisation de chaque soldat. Si on suppose que chacun dispose d'un outil de communication sans fil (un laptop, un PDA et/ou un téléphone portable), l'ensemble de ces outils constitue un réseau sans fil. Les commandants doivent avoir des mémoires additionnelles dans leurs outils pour stocker et maintenir toutes les informations de leurs bataillons. Au sein d'une armée, les bataillons doivent coordonner leurs activités. Il en est de même pour toutes les armées de la coalition.

Dans le domaine de la médecine et pour une branche donnée, il y a un spécialiste, des médecins et des auxiliaires de santé. Tout le monde doit consulter le spécialiste, le médecin a quelques responsabilités alors que l'auxiliaire de santé n'a aucune option de prise de décision. Si tout ce staff doit travailler pour sauver des vies humaines dans une forêt ou un désert, ils doivent toujours échanger des données et souvent en temps réel.

De plus, les spécialistes doivent se concerter pour chaque cas. Tout compte fait, qu'il s'agisse du domaine militaire ou médical, les résultats et le délai pour les obtenir constituent des critères d'évaluation de la réussite d'une opération militaire ou médicale.

Cette réalité observée se retrouve dans les unités mobiles qui peuvent constituer un réseau ad hoc. En effet, un laptop (par exemple pentium III, 1.2 GHz, 30 GB), un PDA et un téléphone portable sont des unités mobiles pouvant toutes échanger des données et assurer la fonction de routage, mais ne disposent pas des mêmes ressources, des mêmes capacités, des même temps de traitement des données. Bref, elles n'ont pas les mêmes potentialités à assumer certaines fonctions. En effet, le téléphone cellulaire, même doté d'option de routage, n'a pas un espace mémoire suffisant pour stocker les informations de routage. En outre, il ne peut pas transmettre, dans les délais convenables, les paquets dont la source et la destination sont des laptops. Ainsi, notre proposition s'inscrit dans une logique qui privilégie les unités mobiles les plus pourvues en potentialités d'agir comme tête de grappe.

3.1.2 Hypothèses

Dans ce mémoire, notre proposition repose sur un certain nombre d'hypothèses de base que nous formulons comme suit :

Dans le réseau ad hoc dense considéré,

- les unités mobiles sont hétérogènes (des laptops avec processeurs, capacités ROM et RAM différentes, des PDA et des téléphones portables avec l'option d'échanges de données) ;
- les unités mobiles sont à l'air libre avec disponibilité du système GPS ;
- les unités mobiles disposent de radios émetteurs-récepteurs équipées de plusieurs éléments d'antennes directionnelles, avec possibilité d'activer ou de désactiver une direction relative à une antenne ;
- les unités mobiles disposent de sources d'énergie durable, en l'occurrence la pile à combustible ;

- les liaisons radio sont fiables (pas d'obstacles, évanouissements ou interférences) et ont un rapport signal à bruit (S/N) acceptable pour toutes les unités mobiles en communications directes. Toute autre communication dont le rapport S/N est inférieur au seuil est considérée comme inexistante.

3.2 Présentation générale de l'architecture

Dans l'architecture de référence présentée au chapitre 2, la grappe est centralisée. Les défauts de la centralisation ont été évoqués. Dans cette section, nous présentons le modèle simplifié de notre architecture, l'initialisation ainsi que les aspects de fiabilité et de disponibilité du service dans cette architecture.

3.2.1 Modèle simplifié de notre architecture

La Figure 3.1 illustre le modèle simplifié de notre architecture. Ainsi, CLHP1 et CLHP2 sont des têtes de grappe principales pour les grappes 1 et 2, tandis que clhs1 et clhs2 sont des têtes de grappe de secours (back up) pour ces mêmes grappes. Que ce soit dans l'adressage hiérarchique ou de base qu'on retrouve dans l'architecture de référence, les routes inter-grappes passent toujours par CLHP-G-CLHP. Cependant, quand le réseau est dense, il existe des passerelles réparties qui peuvent aussi être prises en considération lors de l'établissement des routes. En outre, des passerelles (G) et/ou passerelles réparties (DG) peuvent aussi être dans une zone de portée mutuelle. Il importe alors de prendre en compte tous ces cas. En effet, avoir plus de liens améliore la fiabilité et donne plus de flexibilité quant au choix de liens satisfaisant des contraintes et une métrique donnée.

Dans cette logique, nous construisons une maille entre toutes les passerelles (G) et/ou passerelles réparties (DG) se trouvant dans une même zone de portée mutuelle. Par exemple, si la source S dans la grappe 1 veut transmettre du trafic de moindre effort, la route peut être S -CLHP1-DG1-G3-G2-CLHP2- D , si cette dernière a la plus grande valeur du Temps d'Expiration de la Route (RET: Route Expiration Time), comme on le verra par la suite. Par contre, si on ajoute les contraintes de délai, de la taille des tampons au niveau des unités mobiles intermédiaires ainsi que des contraintes sur la

bande passante que le trafic de la source S exige, la route précédente peut ne plus être choisie. On pourrait avoir par exemple la route S -CLHP1-DG1-DG2-CLHP2- D .

Dans cette architecture, l'optimisation d'une route se base sur la valeur du Temps d'Expiration de la Route (RET). Ce dernier est pris comme métrique principale, mais les contraintes citées ci-haut doivent d'abord être satisfaites.

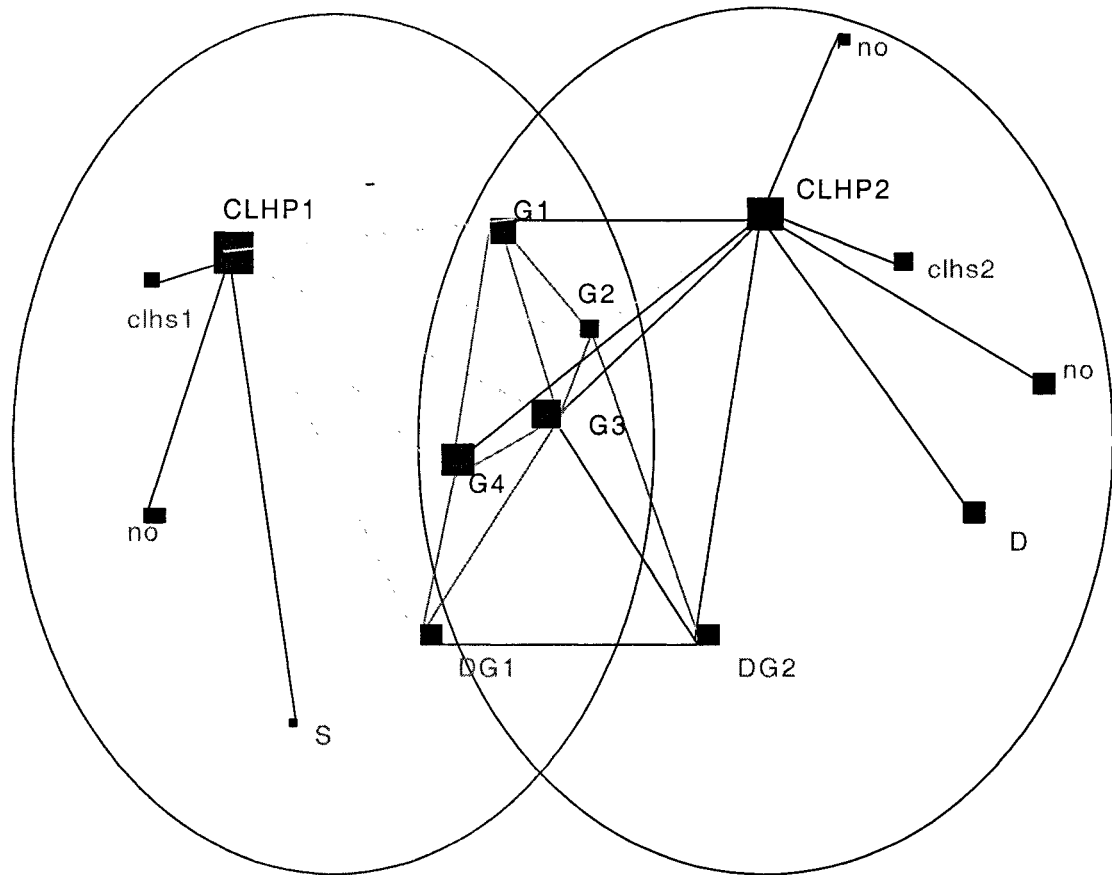


Figure 3.1 Modèle simplifié de l'architecture proposée

3.2.2 Initialisation

Dans nos hypothèses, nous avons supposé que les unités mobiles sont équipées de M éléments d'antennes correspondant à M directions différentes. Comme au départ aucune unité mobile n'a un statut, les unités mobiles doivent activer toutes les directions (canaux des radio émetteurs-récepteurs) et antennes correspondantes, ce qui correspond à un fonctionnement omnidirectionnel de l'émetteur et du récepteur (OROT). Chaque unité mobile diffuse un message de signalisation contenant toutes les informations

reçues du système GPS ainsi que son indice de potentialité à devenir une tête de grappe. L'indice de potentialité tient compte de l'espace disque, du mémoire RAM et de la vitesse du processeur de chaque unité mobile du réseau. Toute unité mobile qui reçoit le message détermine le temps d'expiration d'un lien LET (Link Expiration Time) relatif au seuil de bon fonctionnement de la couche MAC. Elle exécute un algorithme d'élection d'une tête de grappe. Elle envoie par la suite une réponse aux voisins dont les LET sont supérieures ou égales au LET précédemment trouvé. Parmi ces voisins, l'unité mobile non encore affiliée, qui a le meilleur indice de potentialité et le plus grand CET (une moyenne des LET entre une unité mobile et son voisinage), devient une tête de grappe et diffuse à tous ses voisins son statut. Le statut de toute unité mobile qui n'a reçu qu'un seul message provenant d'une tête de grappe est «nœud ordinaire» (no). Elle désactive par la suite $M-1$ directions pour ne garder que la direction et l'antenne relatives à la tête de grappe qui lui a envoyé le message. Son fonctionnement est le DRDT (Réception Directionnelle et Transmission Directionnelle). Par contre, si elle a reçu plus d'un message provenant des têtes de grappe, elle devient une passerelle. Elle diffuse son statut aux autres passerelles (G), aux passerelles réparties (DG) ainsi qu'aux têtes de grappes avec lesquelles elle communique directement. Elle active ainsi les équipements de transmission relatifs à ces derniers. Les unités mobiles dont le statut est tête de grappe laissent toutes les directions et antennes actives. Leur fonctionnement est le OROT (Réception Omnidirectionnelle et Transmission Omnidirectionnelle). Elles enclenchent par la suite les batteries à piles combustibles. Elles conservent par la suite, dans une base de données, toutes les informations des unités mobiles appartenant à leur grappe à savoir leur statut, les LET, les accélérations (angulaire et linéaire) ainsi que les informations fournies par le système GPS. La Figure 3.2 résume cette étape d'initialisation.

3.2.3 Aspects de fiabilité et de disponibilité du service

Dans notre architecture, vu que les unités mobiles sont hétérogènes, toutes les informations de localisation seront stockées dans une base de données au niveau de la tête de grappe, à l'image du HLR en téléphonie cellulaire. Cependant, la tête de grappe

constitue un point critique pour le bon fonctionnement de la grappe et, par conséquent, de toutes les communications dont la grappe sert de transit. Ainsi, il importe de prévoir des mécanismes d'augmentation de la fiabilité et la disponibilité du service. Une des solutions est la réplication.

La réplication est une technique appliquée aux bases de données et aux systèmes répartis pour maintenir automatiquement la disponibilité des données et du service en dépit des défaillances des serveurs. Elle permet d'améliorer le service, d'augmenter les performances et la disponibilité d'un réseau. Dans la littérature, des schémas de réplication comme les schémas actif, passif, semi-actif et semi-passif ont été proposés. Dans tous ces schémas, si N désigne le nombre total d'octets à transférer d'un serveur principal à chacun des M serveurs de secours (backup), alors $N*M$ est la taille totale des messages échangés sur les liens entre le serveur source et les serveurs cibles et inversement (Felber et Schiper, 2001)

Le fait que la bande passante soit limitée nous oriente vers une réplication passive car elle s'approche plus de notre cas. Cependant, le fait que, dans ce schéma, la copie primaire doit diffuser les messages de mise à jour à ses copies secondaires nous limite sur le nombre de copies secondaires à choisir. Ainsi, nous choisissons une seule copie active et la disponibilité sera $d = 1 - p^2$ ou p est la probabilité pour qu'une unité mobile tombe en panne. Pour diminuer au maximum la latence due au processus de reconfiguration (exécution de l'algorithme d'élection d'une nouvelle tête de grappe) en cas de panne de la copie primaire, les unités mobiles seront dès le départ classées dans une liste en fonction de leurs potentialités à agir comme tête de grappe. Ainsi, dès qu'il y a une panne, le suivant de la liste assurera la fonction de son prédécesseur. Deux cas se présentent :

- Le premier, le plus intuitif, est que si la copie primaire tombe en panne, la copie secondaire (le suivant de la liste) prend la relève. Elle envoie ensuite toute sa base de données au suivant de la liste qui devient alors la copie secondaire.
- Si la copie secondaire tombe en panne, le suivant de la liste le remplace et la tête de grappe doit envoyer à ce dernier toute la base de données. Son

implémentation peut se faire à l'aide d'un mécanisme de délai de garde (time-out). Ainsi, si après le délai de garde la copie primaire ne détecte pas la présence de la copie secondaire, elle considère qu'elle est en panne ou éteinte (car le LET n'est pas

Pour toute unité mobile appartenant au réseau :

- Activation de toutes les directions
- Diffusion du message de signalisation
- Si réception des messages de signalisation, élection d'une tête de grappe
- Le statut du nœud élu est CLHP (pour tête de grappe principale)
- Si statut = CLHP, alors
 - fonctionnement OROT
 - diffusion d'un message de confirmation d'élection à tous ses voisins
 - Enclenchement de la pile à combustibles
- Si une unité mobile reçoit un seul message provenant d'une tête de grappe
 - Si elle peut toujours communiquer directement avec une autre unité mobile ayant reçu un message de confirmation d'une tête de grappe différente
 - Son statut = passerelle répartie (DG)
 - Désactivation des directions autres que celles relatives aux DG et à la tête de grappe.
 - Sinon son statut = passerelle (G)
 - Désactivation des directions autres que celles relatives aux têtes de grappe dont elle a reçu des messages
 - Sinon son statut = nœud ordinaire (no)
 - Désactivation des directions autres que celles relatives à la tête de grappe dont elle a reçu le message de signalisation.

Figure 3.2 Procédure d'initialisation

encore atteint) et par la suite elle envoie toute la base de données au nœud qui suit dans la liste. Ce dernier devient la copie secondaire.

3.2.4 Communication entre le CLHP et le clhs

Dans cette section, nous présentons les différents éléments stockés dans les bases de données. Ces dernières sont localisées au niveau des têtes de grappes principales et secondaires (*CLHP* et *clhs*). Les algorithmes de choix du type de mise à jour ainsi que ceux de mise à jour locale et globale sont présentés.

Au niveau des têtes de grappes principales et secondaires sont stockées :

- Une base de données de localisation de toutes les unités mobiles de la grappe; ces éléments sont, pour chaque unité mobile, les vecteurs vitesse et position, ainsi que la direction de déplacement ;
- Une table de statut des unités mobiles de leur grappe (nœud ordinaire no, une passerelle G_{ij} entre deux têtes de grappe i et j , une passerelle répartie DG_{ij} entre grappes contenant les unités mobiles i et j) ;
- Une table de routage inter-grappes qui contient une liste de routes déjà établies. On spécifiera par la suite comment les routes sont établies. Cette table sera transmise à l'étape d'initialisation au *clhs* ou, en cas de panne de ce dernier, au suivant de la liste.

Parmi les données de localisation, on retrouve le vecteur vitesse ayant les coordonnées v_x, v_y, v_z , le vecteur position (x, y, z) et la direction de déplacement $\theta(t)$. La vitesse peut être une constante v_0 ou une fonction linéaire du temps $v(t)$. Le vecteur position dépend de la vitesse et de la direction de déplacement. Si la variable de la vitesse change (i.e si l'accélération $\gamma(t)$ augmente, diminue ou s'annule), le *CLHP* enverra un paquet de mise à jour au *clhs*. En effet, les composantes du vecteur position sont des fonctions de la vitesse, de la direction de déplacement et du temps. La vitesse est aussi une fonction du temps, des accélérations linéaire et angulaire. Ainsi, par exemple, $x = f(v, \theta(t), t)$; $\theta = f(\gamma', t)$; $v = f(\gamma, t)$ où γ est l'accélération linéaire, γ' l'accélération angulaire. Donc, nous voyons que tous les paramètres de localisation

varient en fonction des accélérations linéaire, angulaire et en fonction du temps. En cinématique, l'unité mobile peut faire trois sortes de mouvement, à savoir une translation, une rotation ou une combinaison des deux. Ainsi, tant que γ et/ou γ' n'auront pas changé et ce, dans l'optique de l'optimisation de la bande passante, le *CLHP* n'enverra pas un message de mise à jour au *clhs* pour cette base de donnée. Si c'est la seule variable temps qui change, il y aura une incrémentation automatique dans toutes les bases de données (copie primaire ou secondaire). Si γ et/ou γ' varient, l'incrément de v , x , y , z et de θ ne dépend plus seulement du temps, d'où l'envoi par la tête de grappe d'un message de mise à jour (et ce, seulement pour l'entrée ayant subi la modification) à la tête de grappe de secours (backup).

Nous allons ainsi définir deux fonctions de mise à jour, une locale qui tient compte du temps et une autre globale qui est appelée lorsqu'en plus du temps, une ou les deux accélérations varient. Les algorithmes de choix du type de mise à jour, de mise à jour locale et globale sont repris aux Figures 3.3, 3.4 et 3.5.

3.3 Algorithme d'élection de tête de grappe

Dans cette section, nous présentons la philosophie de conception, les bases ainsi que les détails de l'algorithme d'élection des têtes de grappe que nous proposons.

3.3.1 Philosophie de conception

Le choix d'un nombre optimal de têtes de grappe qui minimise le délai de transmission tout en augmentant le débit reste toujours un problème. Dans cette logique, nous allons proposer un algorithme qui tient compte de l'utilisation des antennes directionnelles et des batteries à autonomie durable (existence de piles à combustibles). Les antennes directionnelles augmentent la portée de transmission, ce qui diminue le nombre de sauts (i.e. le nombre de têtes de grappes de transit), la latence de transmission d'un paquet entre une source et une destination.

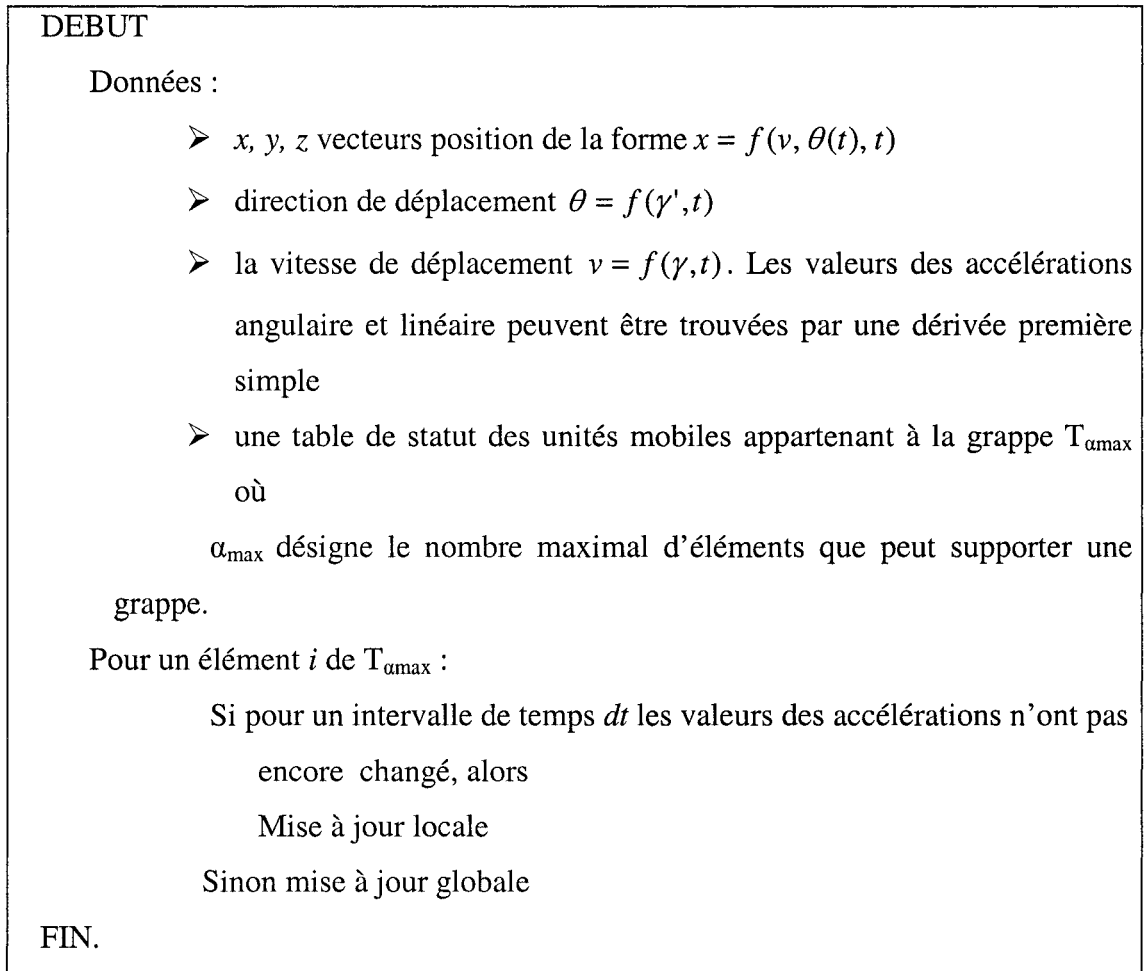


Figure 3.3 Procédure de choix du type de mise à jour

De plus, cela améliore la consommation de la puissance de transmission. En outre, une meilleure portée de transmission permet d'éviter de relancer souvent l'algorithme d'élection de la tête de grappe et donc de formation de grappe. Cet algorithme se base sur les paramètres indiquant les potentialités à devenir des têtes de grappes, en l'occurrence la capacité ROM, la vitesse du processeur et la capacité RAM. En effet, une vitesse élevée du processeur améliore le temps de traitement et donc réduit le temps total de transmission. S'il y a plus d'unités mobiles que le seuil de la couche MAC à un instant donné, certains paquets doivent être mis dans un tampon. Ce dernier est une partie du ROM. La RAM agit également sur le délai

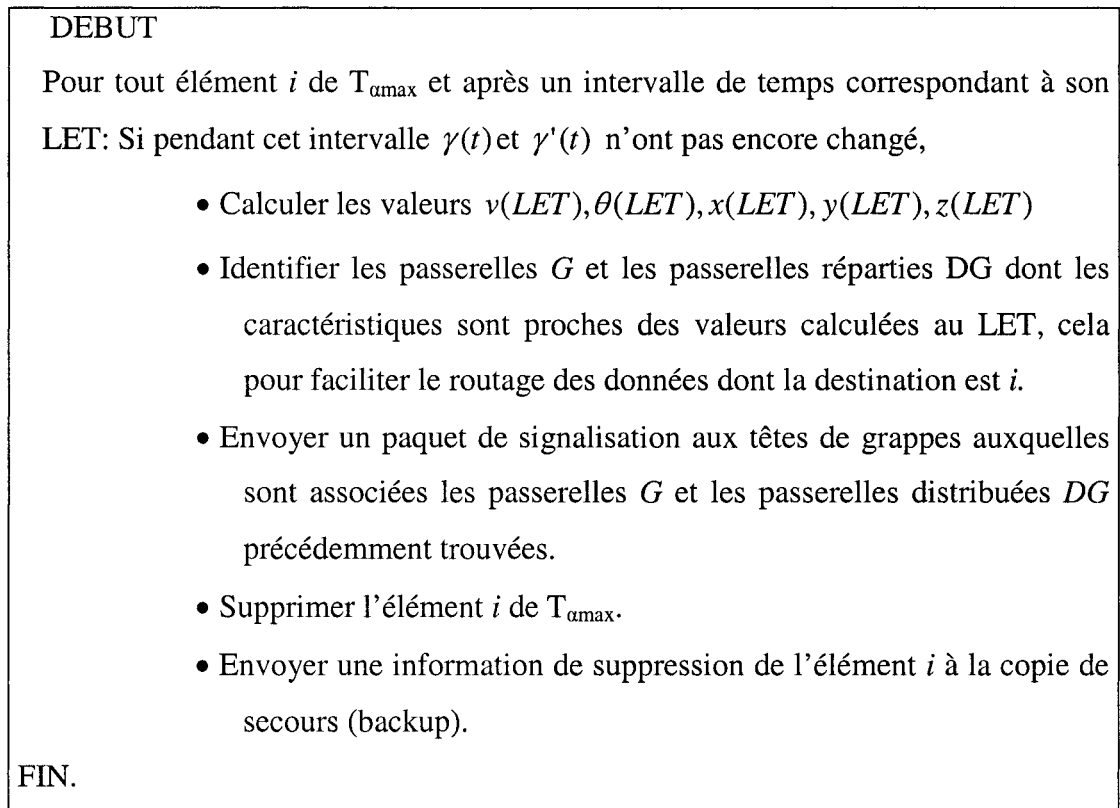


Figure 3.4 Procédure de mise à jour locale

Comme paramètre, on prendra également le temps d'expiration d'un lien (LET : Link Expiration Time) entre une tête de grappe et ses voisins, la mobilité relative ainsi que le domaine de bon fonctionnement de la couche MAC. Un des problèmes de la notion de mise en grappe est la centralisation du réseau. En effet, la tête de grappe devient un point critique du réseau, comme nous l'avons déjà mentionné à la section 3.2.2. Pour cela, les unités mobiles seront, à l'étape d'initialisation, classées par ordre croissant de potentialités à devenir des têtes de grappe.

DEBUT

Pour tout élément i appartenant à $T_{\alpha\max}$,

Si au bout d'un intervalle de temps dt , $\gamma(t)$ et/ou $\gamma'(t)$ changent alors :

- Calculer $v(dt), \theta(dt), x(dt), y(dt), z(dt)$ en utilisant les anciennes valeurs de $\gamma(t)$ et $\gamma'(t)$
- remplacer les anciennes valeurs de $\gamma(t)$ et/ou de $\gamma'(t)$ par les nouvelles
- Si la norme du vecteur position est encore inférieure à la portée de transmission,
 - calculer la nouvelle valeur du LET,
 - envoyer une information de mise à jour (contenant les nouvelles valeurs des accélérations, du LET, des vecteurs positions et de la direction de déplacement) à la copie de secours (back up)

Sinon

- Identifier les passerelles G et les passerelles réparties DG dont les caractéristiques sont proches des valeurs calculées au LET. Cela pour faciliter le routage des données dont la destination est i .
- Envoyer un paquet de signalisation aux têtes de grappes auxquelles sont associées les G et les DG précédemment trouvées
- Supprimer i de $T_{\alpha\max}$

Envoyer une information de suppression de i à la copie de secours (back up).

FIN.

Figure 3.5 Procédure de mise à jour globale

Pour prévenir les conséquences d'une panne quelconque de la tête de grappe, celle-ci sera toujours répliquée à l'unité mobile qui la suit dans la liste. Si on calcule les différents Temps d'Expiration d'un Lien (LET), on se fixe un quorum α_{\min} qui constitue une limite inférieure pour relancer la procédure d'élection et de formation de grappe. En effet, pour chaque unité mobile de la liste, on détermine ses voisins. On évalue le LET de chacun des liens qui le relie à ses voisins. Si, après un temps t , le nombre de liens encore actifs devient inférieur à α_{\min} , on relance l'algorithme d'élection. Cependant, si avant que α_{\min} ne soit atteint, la tête de grappe est déconnectée de sa grappe (panne, arrêt), la tête de grappe de secours (backup) prend la relève et le suivant de la liste devient le backup. En outre, en se fixant seulement la valeur de α_{\min} (relatif au LET), c'est comme si on traitait un seul cas, celui où les unités mobiles s'éloignent de la tête de grappe et sortent de la grappe. Les unités mobiles se déplacent d'une grappe pour entrer dans une autre. Ainsi, les voisins d'une tête de grappe peuvent augmenter. Il est donc nécessaire de se fixer α_{\max} comme limite supérieure de bon fonctionnement de la couche MAC. Au-delà de α_{\max} , l'algorithme d'élection de tête de grappe et de formation de grappe devrait être relancé. Pour l'implémentation, on peut considérer un compteur c_{vtg} de voisins d'une tête de grappe. Si la tête de grappe j a reçu un paquet d'information venant d'une autre grappe i lui indiquant l'arrivée d'une unité mobile, il y a incrémentation de c_{vtg} dans j et une décrémentation de c_{vtg} dans i . D'où un intervalle de bon fonctionnement $[\alpha_{\min}, \alpha_{\max}]$ pour la tête de grappe. D'une façon simplifiée, si $(\alpha_{\min} + \alpha_{\max})/2$ est le nombre de membres qu'une grappe peut couvrir à l'optimalité, on évalue le $LET[(\alpha_{\min} + \alpha_{\max})/2]$ qu'on peut appeler LET_{opt} . À l'expiration du LET, l'unité mobile j sort de la grappe i et entre dans une grappe j .

Ainsi, si $c_{vtgi} \geq \alpha_{\min}$ et $c_{vtgj} \leq \alpha_{\max}$, alors on décrémente le compteur dans la grappe i c_{vtgi} et on l'incrémente pour la grappe j c_{vtgj} . Sinon, si $c_{vtgi} < \alpha_{\min}$, lancer l'algorithme d'élection et de formation de la grappe i .

Si $c_{vtgj} > \alpha_{\max}$, lancer l'algorithme d'élection et de formation de la grappe j . Dans les sections qui suivent, nous présentons les bases de l'algorithme ainsi que l'algorithme d'élection d'une tête de grappe.

3.3.2 Les bases de l'algorithme

L'algorithme que nous proposons repose sur certaines bases que nous énonçons comme suit :

- La procédure d'élection de la tête de grappe n'est exécutée que si les bornes de l'intervalle sont atteintes. La mise à jour globale ne sera faite que si les paramètres fondamentaux de la mobilité (accélération angulaire ou linéaire) varient. La mise à jour locale est une fonction du temps.
- Pour assurer convenablement la tâche de tête de grappe, chaque tête de grappe pourra supporter un nombre de voisins variant dans l'intervalle $[\alpha_{\min}, \alpha_{\max}]$. Si $N(v)$ est inférieur à α_{\min} , il y a un gaspillage de ressource. Par contre, si $N(v)$ est supérieur à la borne supérieure α_{\max} , il y a une congestion au niveau de la file d'attente, ce qui augmente le délai de transmission.
- La mobilité relative est un facteur utilisé pour élire une tête de grappe (McDonald et Znati, 1999). En effet, nous avons supposé une disponibilité des informations de localisation fournies par le système GPS. Cependant, une unité mobile qui se déplace moins vite par rapport à un référentiel associé à ce système peut s'écarter beaucoup plus rapidement de ses voisins. D'où l'importance de considérer la mobilité relative. Nous pouvons l'appliquer comme suit dans le cas de notre problème. Nous trions premièrement les unités mobiles en fonction de l'indice de potentialité. Dans cette liste triée, nous évaluons pour chaque élément de la liste la mobilité totale relative (Δ_v). Si Δ_{vij} est la mobilité relative à l'unité mobile j par rapport à l'unité mobile i , alors Δ_v de l'unité mobile i est la somme des Δ_{vij} pour tout j appartenant à $N(i)$ et est donnée par la formule suivante :

$$\Delta_v = \sum_{j \in N(i)} \Delta_{vij}.$$

On peut choisir l'élément qui a une valeur de Δ_v plus petite et dont l'indice Pot est élevé. Cependant, la formule de calcul du LET est complète car elle considère les valeurs relatives des paramètres de deux unités mobiles en communication. Le choix de la tête de grappe principale peut se faire en considérant seulement les valeurs du LET au seuil qu'on peut appeler CET (Cluster Expiration Time). La mobilité relative, combinée à la direction de déplacement relative et du LET, pourra guider le routage des communications dont la destination est l'unité mobile qui vient de sortir de la grappe.

3.3.3 Algorithme d'élection d'une tête de grappe

Dans cette section, nous présentons l'algorithme d'élection d'une tête de grappe. Cet algorithme comprend quatre étapes :

Étape 1 : Détermination des indices de potentialités.

On trie les unités mobiles en fonction des potentialités à assurer correctement la fonction de tête de grappe. Les paramètres qui déterminent cette potentialité sont : la vitesse du processeur $v_{process}$, la capacité de la mémoire ROM m_{rom} et la capacité de la mémoire RAM m_{ram} . On associe à ces paramètres des coefficients qui indiquent le poids de chaque paramètre; soit $\alpha_1, \alpha_2, \alpha_3$ tels que : $\alpha_1 + \alpha_2 + \alpha_3$ respectivement pour la vitesse du processeur ($v_{process}$), la mémoire ROM (m_{rom}) et la mémoire RAM (m_{ram}). Les valeurs de ces coefficients dépendent des applications que va supporter le réseau. En effet, si le réseau va supporter une application temps réel, $\alpha_1 \geq \alpha_3 \geq \alpha_2$. Pour les applications du moindre effort, tant qu'il n'y a pas de congestion, c'est la mémoire ROM qui importe: $\alpha_2 \geq \alpha_3 \geq \alpha_1$.

Ainsi, dans chaque cas (application temps réel ou meilleur effort), on calcule l'indice de potentialité par la formule :

$$\text{Indice_pot} = \alpha_1 * m_{rom} + \alpha_2 * m_{ram} + \alpha_3 * v_{process}.$$

Ces unités mobiles sont classées par ordre croissant des indices de potentialité (Indice_Pot) et sont conservées dans une liste.

Étape 2 : Pour chaque élément de la liste, calculer le nombre de voisins.

$$N(v) = \{v' \neq v / d(v, v') < tx_range\}$$

Étape 3 :

Pour chaque unité mobile de la liste et pour chaque élément de son voisinage, calculer le LET, ensuite déterminer le LET correspondant à un seuil qu'on peut fixer à $(\alpha_{min} + \alpha_{max})/2$ pris comme quorum exigible pour un fonctionnement (sans gaspillage de ressource ni surcharge de la file d'attente) acceptable de la couche MAC. On peut appeler ce LET au seuil, un CET (Cluster Expiration Time). Par simplification, on peut considérer comme CET d'une unité mobile la moyenne des LET.

Étape 4 :

On définit un indice d'élection (indice_election) pour chaque unité mobile. Cet indice d'élection est une combinaison linéaire de l'indice de potentialité et du CET (qui correspond à l'indice de stabilité de la grappe). On choisira comme tête de grappe principale l'unité mobile qui a l'indice d'élection (indice_election) le plus élevé, comme l'illustre la Figure 3.6.

Si *CLHPi* est élue, tous ses voisins limités en nombre par la valeur de α_{min} ne peuvent plus être élus car ils sont déjà affiliés à une grappe. Pour choisir la tête de grappe de secours (backup), on doit considérer la mobilité relative (vitesse relative). Les unités mobiles affiliées à une grappe sont classées suivant la vitesse relative au référentiel associé à la tête de grappe et l'indice de potentialité (indice_Pot). La meilleure est choisie comme tête de grappe secondaire (*CLHS*), les autres unités mobiles affiliées à la grappe sont des nœuds ordinaires, des passerelles ou des passerelles réparties. Tant que toutes les unités mobiles de la liste ne sont pas affiliées à une grappe, on répète l'algorithme.

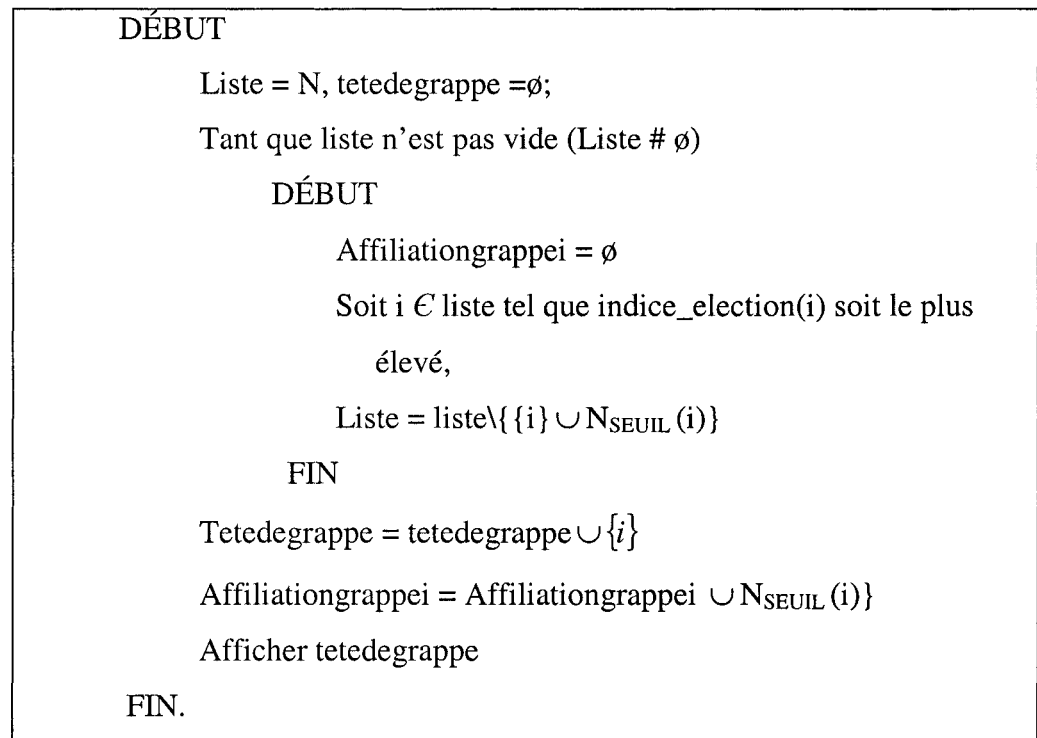


Figure 3.6 Algorithme d'élection d'une tête de grappe

3.4 Routage inter-grappes

Dans le premier chapitre, nous avons montré l'avantage de la méthode de prédiction de mobilité. La stabilité d'une route est la métrique de routage. Pour d'autres algorithmes de routage par exemple *multicast*, le délai est la métrique privilégiée. Cependant, le réseau n'est généralement pas limité aux applications et services qu'il peut rendre. Ainsi, il est nécessaire d'inclure à ces protocoles une qualité de service. En effet, pour les applications temps réel, en plus de la stabilité d'une route, il y a des contraintes de délai et de bande passante qui doivent être respectées. Pour les applications du meilleur effort, la fiabilité est la seule contrainte.

3.4.1 Choix d'un protocole adapté au routage inter-grappes

Dans cette section, nous choisissons, parmi les protocoles de routage *multicast* déjà élaborés dans les réseaux ad hoc, celui qui peut convenir à notre problème. C'est à ce protocole que nous allons appliquer la méthode de prédiction de la mobilité et les différentes contraintes. Ce choix peut ne pas être optimal vu qu'on continue à développer d'autres protocoles. Cependant, cette méthode et ces contraintes peuvent s'appliquer à plusieurs protocoles. Le protocole choisi sera adapté aux types de service offert par le trafic qui arrive (trafic *temps réel* et trafic du *moindre effort*). Les critères de sélection des routes sont : le délai, le RET, la bande passante ainsi que la capacité des tampons au niveau des unités mobiles de transit (vu que le réseau est hétérogène) pour les applications temps réel, tandis que le RET peut à lui seul constituer une métrique pour les applications du moindre effort.

Dans (Lee et al., 2000), les performances de cinq protocoles de routage *multicast* applicables aux réseaux ad hoc, en l'occurrence AMRoute, AMRIS, CAMP, ODMRP ainsi que la diffusion (flooding), ont été comparées. Les métriques de comparaison sont : le rapport de transmission des paquets, le rapport entre le nombre de paquets transmis au nombre de paquets reçus, le nombre de paquets de contrôle transmis par octet de données effectivement reçues, ainsi que le rapport entre le nombre de paquets de contrôle et de données transmis par paquets reçus. Toutes ces métriques ont été évaluées en fonction de la mobilité, du nombre de sources, de la taille du groupe de diffusion (*multicast*) ainsi que de la charge du trafic.

Dans notre problème, les sources *multicast* sont des têtes de grappe et sont limitées en nombre. Les membres du groupe *multicast* sont des passerelles (*G*) et des passerelles réparties (*DG*). Leur nombre varie (augmente ou diminue) en fonction de la mobilité des unités mobiles. En outre, nous allons utiliser la méthode de prédiction de la mobilité, il n'y aura pas de surcharge due aux paquets de contrôle. Cependant, la charge peut augmenter si la majorité des communications, à un instant *t*, sont inter-grappes. Dans la plupart des cas, l'évaluation des métriques citées ci-haut montre que ODMRP a de bonnes performances par rapport aux autres. Le protocole ODMRP peut donc être

choisi pour le cas de notre problème. Ainsi, dans la suite, nous appliquons les contraintes et la méthode de prédiction de la mobilité à ce protocole.

3.4.2 Application du mécanisme de prédiction de la mobilité avec contraintes

Dans cette section, nous montrons les métriques utilisées pour les différentes variantes du protocole ODMRP, les mécanismes de sélection de route ainsi que les procédures de détermination du type de trafic et de routage inter-grappes. Dans ODMRP sans contraintes ni prédiction de la mobilité, les routes sont choisies en utilisant comme métrique le délai (Su et al., 2001; Lee et al., 1998; Gerla et al., 2000). Dans la version de ODMRP qui intègre la prédiction de la mobilité, la métrique utilisée est la stabilité de la route (i.e. la valeur du RET la plus grande) (Lee et Gerla, 1999). Cette métrique peut aussi s'appliquer au trafic du meilleur effort dans le cas de notre problème. Quant au trafic qui exige une qualité de service, nous allons ajouter les contraintes de bande passante et de capacité de tampon au niveau de l'unité mobile de transit, ainsi que la limite sur le délai de transmission entre la source et la destination.

Chaque unité mobile choisit, parmi les liens qui lui sont adjacents (relatifs à une destination donnée), ceux qui respectent les contraintes précitées. Elle envoie sur ces derniers le paquet de donnée. Il revient à la destination de choisir la meilleure route. Ainsi, on ajoute seulement, au paquet incluant le mécanisme de prédiction de la mobilité, un champ supplémentaire indiquant les spécifications sur la qualité de service, en l'occurrence la bande passante nécessaire, la limite sur le délai de transmission ainsi que la capacité de la mémoire tampon exigible.

Chaque unité mobile de transit (qui peut être une tête de grappe principale *CLHP*, une passerelle *G* ou une passerelle répartie *DG*) maintient une table des voisins, les caractéristiques des liens (délai de transmission, bande passante résiduelle) qui lui sont adjacents, ainsi que les tailles des tampons des unités mobiles voisines servant de transit. Le RET est calculé à la demande. La source *multicast* (dans ODMRP), qui est dans notre cas la tête de grappe (CLHP->S) abritant la source, envoie un paquet « JOIN DATA » auquel elle ajoute le LET (S- (CLHP->S)), sélectionne parmi les liens qui le relient à ses

voisins, relatifs à la destination (qui sont des passerelles G et passerelles réparties DG), ceux qui respectent les contraintes de bande passante et de capacité. Elle ne peut pas évaluer le délai car c'est un paramètre cumulatif. Elle vérifie seulement si le délai cumulé sur les liens précédents et le lien adjacent ne dépasse pas la valeur de la contrainte de délai.

Le minimum entre le LET reçu et celui calculé, le délai cumulatif, ainsi que le minimum entre la bande passante disponible entre le lien en amont et en aval sont appliqués dans les champs appropriés. De la même façon qu'en ODMRP avec prédiction de la mobilité sans contraintes, l'unité mobile courante applique dans un champ approprié sa propre localisation, sa vitesse ainsi que sa direction de déplacement avant de transférer le paquet. Quand le paquet arrive à la tête de grappe où est localisée la destination D ($CLHP < -D$), la tête de grappe sélectionne la route la plus stable, i.e. celle qui a la plus grande valeur du RET. Au cas où deux meilleures routes ont la même valeur du RET, celle qui a le délai minimal est privilégiée. En cas d'égalité des deux métriques, le nombre de sauts devient la métrique de sélection de la meilleure route. Les procédures de détermination du type de trafic et du routage inter-grappes sont illustrées aux Figures 3.7 et 3.8.

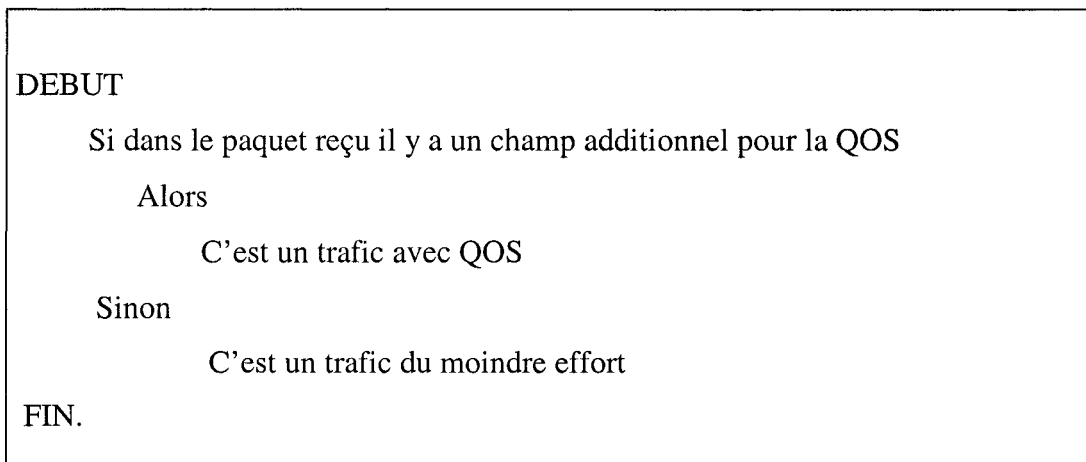


Figure 3.7 Procédure de détermination du type de trafic

3.4.3 Détermination de la meilleure route

Dans cette section, nous présentons en détail la procédure de sélection de route. En nous référant à notre architecture, supposons qu'un nœud ordinaire no dans une grappe i veut transmettre du trafic à un autre nœud ordinaire dans une grappe j ($i \neq j$). Ce trafic est décrit par le couple (b, r) , B et d , où b est la taille du paquet, r le taux d'arrivée du trafic, B la bande passante et d la limite sur le délai de transmission. Le trafic doit nécessairement passer par les têtes de grappe i et j . Au niveau de la destination, comme la probabilité pour que le nœud ordinaire soit limité en ressources (nécessaires aux calculs et comparaisons) est élevée, la tête de grappe j à laquelle est affilié la destination, choisira la meilleure route. Ainsi, la tête de grappe abritant la source ($clhp \leftarrow S$) diffuse le paquet de trafic aux passerelles (G) et aux passerelles réparties (DG) relatives à la destination (identifiées par la localisation de la destination). Chaque unité mobile de transit ajoute au paquet de requête de route, le LET, la bande passante résiduelle B_r , ainsi que le délai de transmission D_r . Le LET et le B_r transmis seront toujours les minima des LET et des B_r entre les liens en amont et en aval d'une unité mobile de transit. Le D_r sera la somme des délais de transmission sur les liens que traverse le paquet et D_r doit être inférieur à d . La taille du tampon à l'unité mobile suivante doit être supérieure à celle indiquée dans le champ réservé à la QOS.

Au niveau de la tête de grappe j où se trouve la destination, le $clhp \leftarrow D$ compare toutes les routes reçues, qui doivent nécessairement remplir (la sélection se faisant de proche en proche) les conditions suivantes :

$$\{min(B_{ri})\} \geq B \text{ et } \sum_{i=0}^{i=n} D_{ri} \leq d$$

On choisit par la suite celle qui a une valeur plus grande du RET. Ainsi, l'optimisation d'une route se fait au niveau de la maille reliant la tête de grappe abritant la source et celle abritant la destination. La procédure de sélection de route est détaillée à la Figure 3.9.

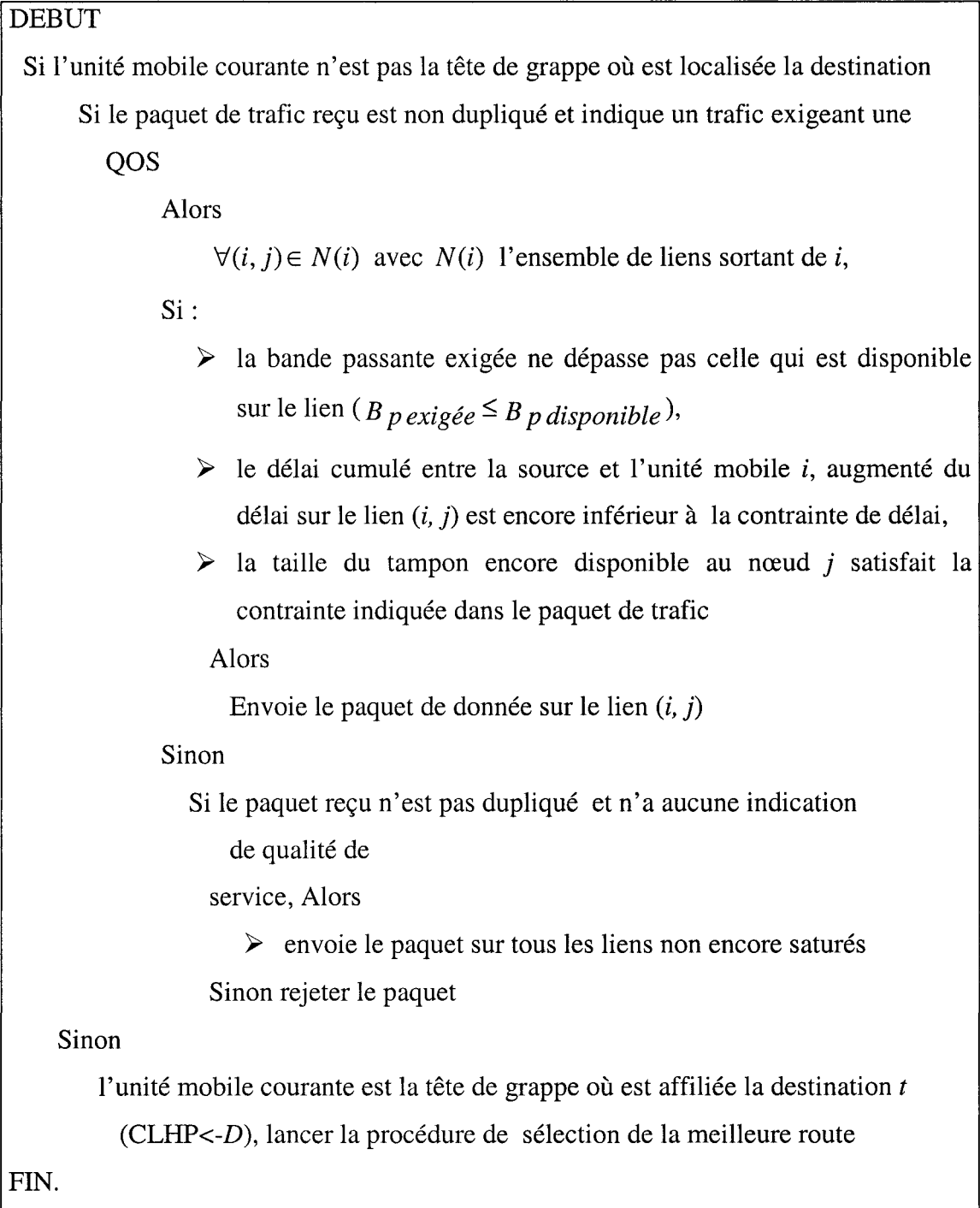


Figure 3.8 Algorithme de routage inter-grappes

DEBUT

Soit N l'ensemble de paquets qui sont arrivés au CLHP<-D, c'est à dire l'ensemble des chemins (respectant les contraintes) entre la tête de grappe où est affiliée la source CLHP<-S et la tête de grappe où est affiliée la destination CLHP<-D,

Si une seule route a une valeur du RET élevée,
cette route est choisie,

Sinon (deux ou plusieurs routes ont la même valeur du RET)

S'il n'y a pas de routes qui ont le même délai,
celle qui a un délai minimum est choisie,

Sinon celle qui a le plus petit nombre de saut est choisie.

Le CLHP<-D envoie au CLHP<-D un paquet indiquant le chemin sélectionné

FIN.

Figure 3.9 Procédure de détermination de la meilleure route

3.4.4 Exemple illustrant le routage inter-grappes

Pour l'exemple de la Figure 3.10, on indique au Tableau 3.1 sur chaque lien le délai, le temps d'expiration du lien, la bande passante résiduelle ainsi que la capacité du tampon disponible au niveau de chaque unité mobile. La source S veut envoyer du trafic qui exige un délai de transmission n'excédant pas 12 sec, une bande passante de 3 unités et une capacité du tampon de deux unités au minimum. Dans l'exécution de la méthode, on commence à supprimer toutes les unités mobiles qui n'ont pas une capacité de stockage de deux unités, ainsi que tous les liens entrant ou sortant de l'unité mobile. Ainsi, dans notre exemple, l'unité mobile $G2$, dont la capacité est 1, est éliminée ainsi que les liens CLHP1-G2, G2-CLHP2, G2-DG2, G2-G3. Ensuite, nous enlevons les liens dont la bande passante est inférieure à 3, soit CLHP1-G4, CLHP1-G3, G4-DG1 et G1-G4. La contrainte de délai étant cumulative, le CLHP2 supprime toutes les routes dont le

délai excède la contrainte diminuée du délai de transmission sur le lien qui le relie à la destination. Par exemple, la route $S\text{-CLHP1-G1-G3-DG2-CLHP2-D}$ (2-1-4-4-4-2) excède 12 et donc jusqu'au nœud de décision CLHP2, le délai dépasse 10 : la route est ignorée. De la même façon, la route $S\text{-CLHP1-DG1-G3-DG2-CLHP2-D}$ (2-2-3-4-2) est exclue. On va ainsi sélectionner, parmi les routes qui restent, celle qui a la meilleure valeur du RET : $S\text{-CLHP1-G1-CLHP2-D}$ (4-5-6-5) $\rightarrow \text{RET} = 4$

$S\text{-CLHP1-G1-G3-CLHP2-D}$ (4-5-4-3-5) $\rightarrow \text{RET} = 3$

$S\text{-CLHP1-DG1-DG2-CLHP2-D}$ (4-5-4-4-5) $\rightarrow \text{RET} = 4$

$S\text{-CLHP1-DG1-G3-CLHP2-D}$ (4-5-2-3-5) $\rightarrow \text{RET} = 2$

Les deux meilleures routes ont une valeur du RET de 4, celle qui a le meilleur délai de transmission est $S\text{-CLHP1-G1-CLHP2-D}$ (2-1-3) \rightarrow délai de 6, alors que l'autre chemin a un délai de 7. Si c'était du trafic meilleur effort, les routes $S\text{-CLHP1-G1-CLHP2-D}$ et $S\text{-CLHP1-G3-CLHP2-D}$ auraient les mêmes valeurs des RET (5) mais $S\text{-CLHP1-G1-CLHP2-D}$ a moins de sauts et c'est elle qui devrait être choisie.

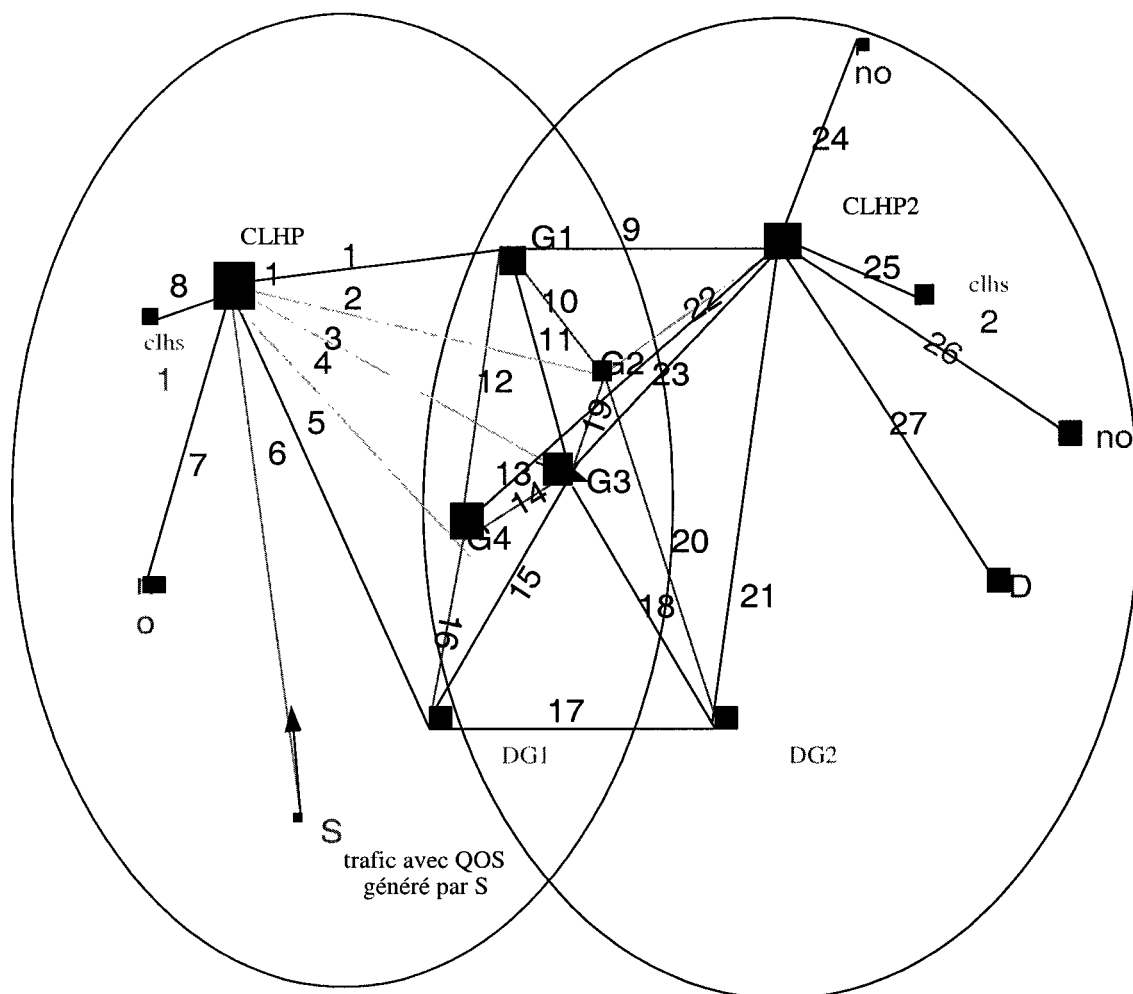


Figure 3.10 Méthode de prédiction de la mobilité avec contraintes

Tableau 3.1 Caractéristiques des liens de la Figure 3.10

Numéro	Lien (i,j) correspondant	Caractéristiques : (Délai, LET, Br)	Taille du tampon au nœud j
1	CLHP1-G1	(1,5,3)	4
2	CLHP1-G2	(4,2,4)	1
3	CLHP1-G3	(2,5,2)	3
4	CLHP1-G4	(1,5,1)	4
5	CLHP1-DG1	(2,5,4)	3
6	S-CLHP1	(2,4,2)	10
7	CLHP1-no	-	-
8	CLHP1-clhs1	(0,5,8,10)	10
9	G1-CLHP2	(3,3,6)	1
10	G1-G2	(2,2,3)	3
11	G1-G3	(4,2,3)	4
12	G1-G4	(3,2,2)	4
13	G4-CLHP2	(4,4,4)	10
14	G4-G3	(4,5,3)	3
15	DG1-G4	(2,2,2)	4
16	DG1-G3	(3,2,4)	3
17	DG1-DG2	(1,4,4)	3
18	G3-DG2	(4,4,4)	3
19	DG2-G1	(3,3,3)	1
20	DG2-CLHP2	(2,4,4)	10
21	G3-CLHP2	(3,2,2)	10
22	G2-CLHP2	(4,3,2)	10
23	G4-CLHP2	(3,2,2)	10
24	CLHP2-no	-	-
25	CLHP2-clhs2	(0,5,8,10)	10
26	CLHP2-no	-	-
27	CLHP2-D	(2,5,5)	3

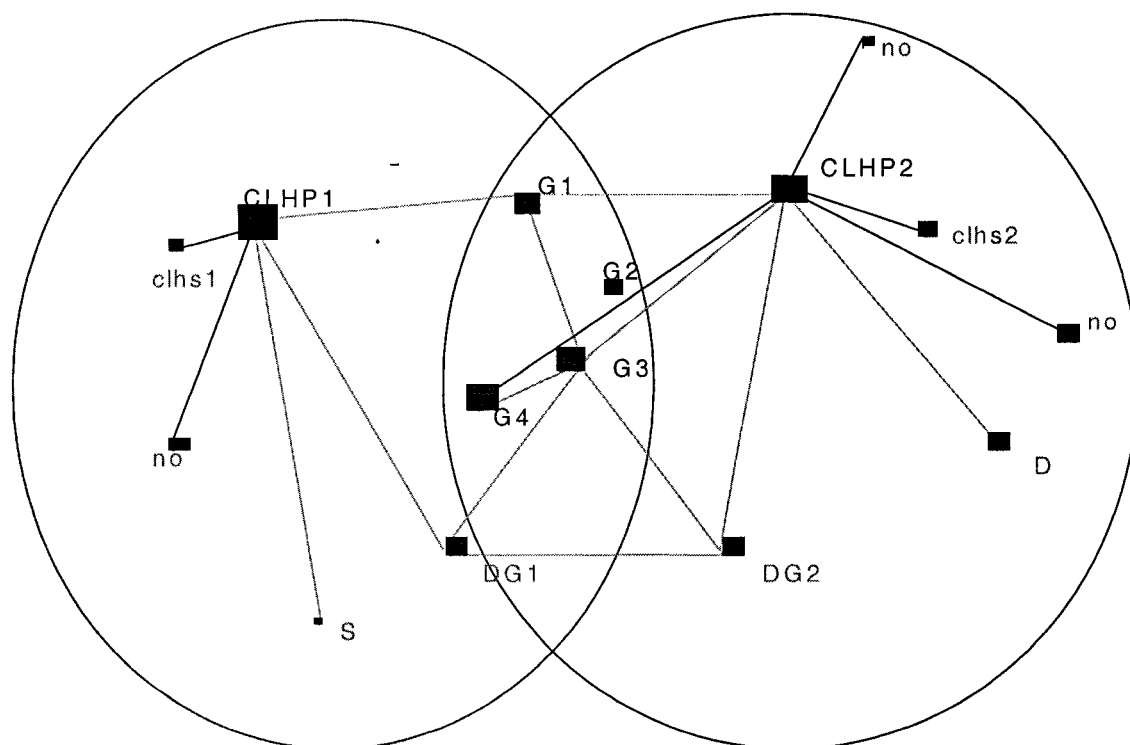


Figure 3.11 Illustration du modèle après suppression des nœuds et des liens ne respectant pas les contraintes

CHAPITRE IV

IMPLEMENTATION ET ANALYSE DES RÉSULTATS

Dans le chapitre précédent, nous avons développé un algorithme d'élection des têtes de grappe et nous avons proposé une amélioration d'une architecture de référence. Ce chapitre expose les détails d'implémentation, les outils utilisés ainsi qu'une analyse des résultats obtenus. Cette implémentation comprend deux parties. Dans la première, nous montrons l'importance de l'algorithme d'élection proposé. Nous le comparons par la suite à l'algorithme WCA qui prend en considération plusieurs paramètres dans le choix d'une tête de grappe. Dans un second temps, nous utilisons le simulateur GloMoSim pour analyser l'impact de l'ajout des contraintes de qualité de service sur les performances du protocole ODMRP (On-Demand Multicast Routing Protocol) avec prédiction de la mobilité.

4.1 Implémentation de l'algorithme d'élection proposé

Dans l'algorithme d'élection que nous proposons, chaque unité mobile diffuse, à l'initialisation, son identificateur *id*, sa position (sur l'axe des *X* et l'axe des *Y*), sa vitesse (*speed*) ainsi que son indice de potentialité (*indice_pot*). L'indice de potentialité est dès le départ calculé par chaque unité mobile en fonction de ses caractéristiques physiques (MROM, MRAM, V-PROCESSEUR) et du type de trafic (temps réel ou meilleur effort). Dans notre implémentation, ces éléments sont lus dans des fichiers que nous avons appelés *test.txt* et *coefficient.txt*.

4.1.1 Principe de fonctionnement et paramètres de base

Dès qu'une unité mobile reçoit un message, elle le stocke dans une table à une entrée correspondant à l'identificateur de l'unité mobile qui l'a envoyé. Elle calcule le LET pour cet élément ainsi que pour l'ensemble du voisinage. Elle calcule par la suite le CET (Cluster Expiration Time) qui peut être considéré comme une moyenne des LET (avec son voisinage) et donc la durée de vie de la grappe formée. Enfin, elle détermine

l'*indice d'élection* qui est une combinaison linéaire de l'*indice de potentialité* et du *CET*. La détermination de l'indice de potentialité est donnée par la formule (4.1).

$$indice_{pot} = \alpha_1 * m_{rom} + \alpha_2 * m_{ram} + \alpha_3 * V_{processeur} \quad (4.1)$$

où $\alpha_1 + \alpha_2 + \alpha_3 = 1$ et $\alpha_1 \geq \alpha_3$ dans le cas du trafic de meilleur effort et l'inverse dans le cas contraire.

L'unité mobile diffuse cet *indice d'élection*, son identificateur (*Id*) ainsi que son statut. Une unité mobile qui reçoit des messages indiquant des indices d'élection plus petits que le sien, se considère comme tête de grappe et son statut devient *CLHP*. Elle envoie alors un message d'élection à ses voisins. Toute unité mobile qui reçoit ce message change de statut et devient *NO* si c'est le premier message reçu; sinon, elle devient une passerelle (*G*).

$$LET = \frac{-(a*b + c*d) + \sqrt{(a^2 + c^2)r^2 - (a*d - b*c)^2}}{a^2 + c^2} \quad (4.2)$$

avec

$$a = v_i \cos \theta_i - v_j \cos \theta_j, \quad (4.3)$$

$$b = x_i - x_j, \quad (4.4)$$

$$c = v_i \sin \theta_i - v_j \sin \theta_j, \quad (4.5)$$

$$d = y_i - y_j, \quad (4.6)$$

$$CET = \frac{1}{n} \sum_{i=0, i \in N(v)}^n LET (tête de grappe et le noeud i) \quad (4.7)$$

Nous avons défini l'*indice de stabilité* (désigné par le *CET*) comme étant une moyenne des *LET* entre une tête de grappe et les membres de la grappe formée. La valeur du *LET* est déterminée à l'aide de la formule (4.2) où *r* désigne la portée de

transmission et les valeurs de a , b , c , d sont données par les formules (4.3), (4.4), (4.5) et (4.6). L'évaluation du CET est illustrée par la formule (4.7).

Avoir un indice de potentialité élevé et assurer la stabilité de la grappe sont deux éléments indépendants. Trouver l'optimum peut être une tâche difficile. Dans notre implémentation, nous avons choisi d'utiliser deux coefficients, l'un pour l'indice de potentialité noté *coeff_indicepot* et l'autre noté *coeff_CET* pour l'indice de stabilité (*CET*). Les valeurs de ces éléments dépendront de l'importance de chaque facteur dans le système considéré. Dans notre simulation, nous avons choisi 0.95 et 0.05 respectivement pour *coeff_indicepot* et *coeff_CET*. Ainsi, l'indice d'élection est donné par la formule (4.8) :

$$Indice_election = coeff_indicepot * indice_pot + coeff_CET * CET \quad (4.8)$$

4.1.2 Détermination de l'indice de potentialité

L'indice de potentialité tient compte des caractéristiques physiques de chaque élément du réseau. Les éléments sont hétérogènes (comme nous l'avons supposé) et, même au sein d'une catégorie, les éléments n'ont pas les mêmes caractéristiques. Du côté des ordinateurs portables, on arrive environ à 1.5 GHz comme vitesse de processeur, 512 méga-octets de mémoire RAM et plus de 80 giga-octets d'espace disque. Nous allons considérer ces éléments dans le réseau de test pour valider notre proposition. Dans notre implémentation, nous supposons que chaque unité mobile récupère ses caractéristiques (mémoire RAM, espace disque, vitesse processeur) dans un fichier. La détermination de l'indice de potentialité tient compte du type de trafic dans le réseau. Les coefficients considérés dans notre implémentation sont lus dans un fichier nommé *coefficient.txt* et y sont stockés sous forme du Tableau 4.1. Pour le trafic temps réel, la vitesse du processeur doit être grande afin de diminuer le temps de traitement au niveau des unités mobiles entre une origine et une destination, et par conséquent, le délai de transmission de bout en bout. Par contre, la fiabilité est privilégiée pour le trafic de meilleur effort, ce qui justifie nos choix.

Dans notre implémentation, nous déterminons une *classe nœud* et comme attributs, nous définissons *capacite_ROM*, *memoire_RAM*, *v_processeur* et *indice_pot*. La valeur de ce dernier est déterminée à l'aide de la formule (4.1), alors que les autres valeurs sont lues dans le fichier *coefficient.txt*.

Tableau 4.1 Coefficients d'indice de potentialité en fonction du type de trafic

Type de trafic	Coeff_MROM	Coeff_MRAM	Coeff_VITESSE_PROCESSEUR
Milleur effort	0.5	0.3	0.2
Temps réel	0.2	0.3	0.5

4.1.3 Détermination du CET

Le CET est une moyenne des LET entre une tête de grappe et ses voisins. Pour chaque unité mobile, les variables d'états x , y , $speed$, respectivement pour les positions sur les axes des X , des Y et la vitesse de déplacement sont définis. Nous définissons deux fonctions *Calcul_LET()* et *calcul_CET()* qui nous retournent respectivement les valeurs du LET et du CET. Par souci de simplification, des attributs *LET*, *CET* conservent ces valeurs pour chaque unité mobile. La valeur du LET est déterminée à partir de la formule (4.2). Nous avons utilisé deux valeurs de seuil que nous avons prédéfinies *Seuil_min* et *Seuil_max*. Ces deux valeurs nous permettent d'une part de favoriser l'élection d'une unité mobile dont le nombre de voisins est compris entre ces dernières, et d'autre part d'optimiser la bande passante.

4.1.4 Les classes utilisées pour implémenter notre algorithme

Nous présentons dans cette section les classes utilisées pour implémenter l'algorithme proposé. Comme nous venons de le montrer dans les sections précédentes, la classe principale est la classe *nœud* (). La classe *Qos* () est utilisée pour distinguer le trafic temps réel du trafic de meilleur effort.

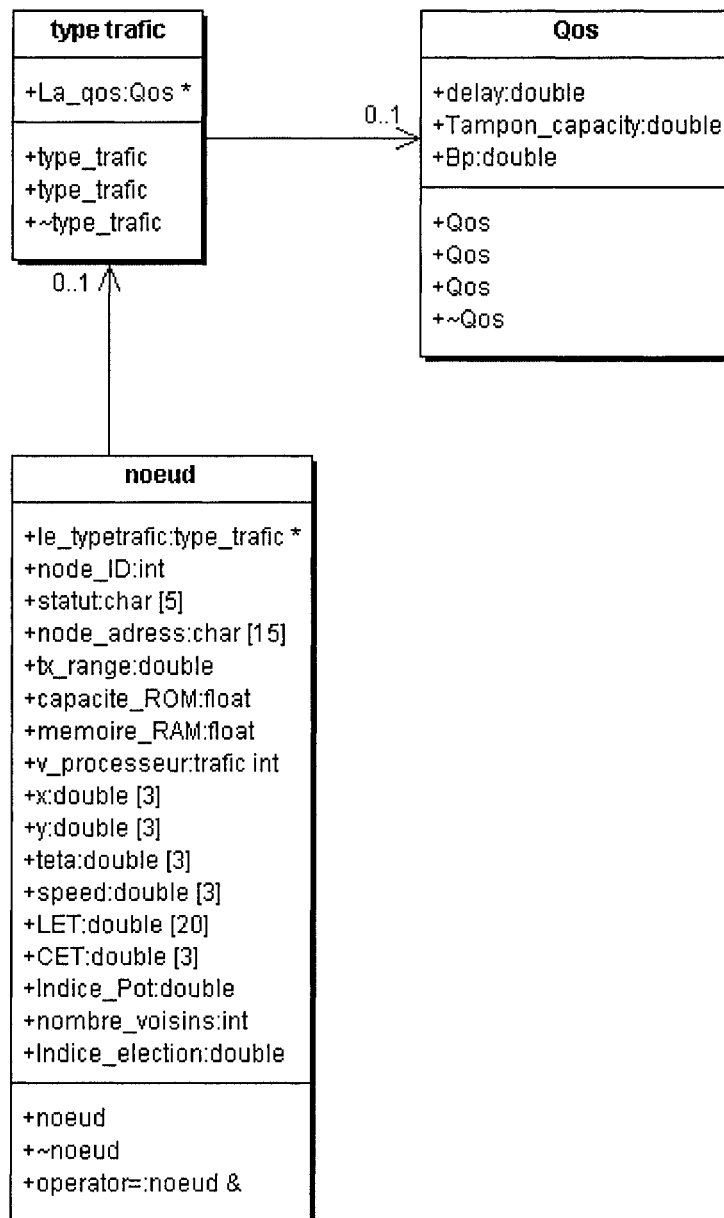


Figure 4.1 Diagramme des classes

4.2 Évaluation des performances

Dans cette section, nous décrivons les métriques de l'évaluation des performances. Un élément du réseau peut être une tête de grappe ou affiliée à une grappe. Trois métriques sont généralement utilisées pour évaluer les performances. Il

s'agit du *nombre de re-affiliations*, du *nombre de têtes de grappe* et du *nombre de mises à jour*.

- Une *re-affiliation* a lieu si une unité mobile change de tête de grappe à laquelle elle est affiliée dans le même ensemble existant de têtes de grappes. Par exemple, si une unité mobile *C* qui était affiliée à une grappe *A*, entre dans une grappe *B*, le mécanisme de *re-affiliation* survient. *C* enclenche le mécanisme d'enregistrement pour détecter la nouvelle tête de grappe, ce qui augmente les messages de signalisation et consomme la bande passante.
- Le *nombre de têtes de grappe* est la cardinalité de l'ensemble dominant des unités mobiles (c'est le nombre d'unités mobiles dont le statut est *CLHP*). Cette métrique constitue un élément important d'analyse. En effet, un nombre optimal de têtes de grappe améliore, par exemple, le délai de transmission de bout en bout (end-to-end delay).
- Si l'unité mobile va dans une région non couverte par une tête de grappe, l'algorithme d'élection est invoqué et un nouvel ensemble de têtes de grappe est obtenu. Si de nouvelles têtes de grappes sont élues, il y a *mise à jour*.

Ces métriques sont analysées en variant le nombre d'unités mobiles dans le réseau, la portée de transmission ainsi que le déplacement maximal.

4.3 Évaluation des algorithmes et des résultats

Pour évaluer les deux algorithmes, nous partons d'une topologie initiale que nous générons aléatoirement. Les unités mobiles peuvent disposer des mêmes types d'interface de communication, ce qui leur donne une même portée de transmission. Dans le cas contraire, les portées de transmission diffèrent. Par simplification, nous considérons le nombre de têtes de grappe comme métrique pour évaluer les

performances de notre proposition. En effet, les deux autres nécessitent un suivi du mouvement des éléments du réseau.

Pour valider et évaluer l'algorithme de référence WCA, nous utilisons une topologie initiale de 15 unités mobiles. Nous nous servons des valeurs de P_v que nous retrouvons dans (Chatterjee et al., 2002). Les valeurs retenues des coefficients sont également prises dans cette référence : $w_1 = 0.7$, $w_2 = 0.2$, $w_3 = 0.05$ et $w_4 = 0.05$. Ces coefficients sont choisis de telle sorte que la somme soit égale à 1.

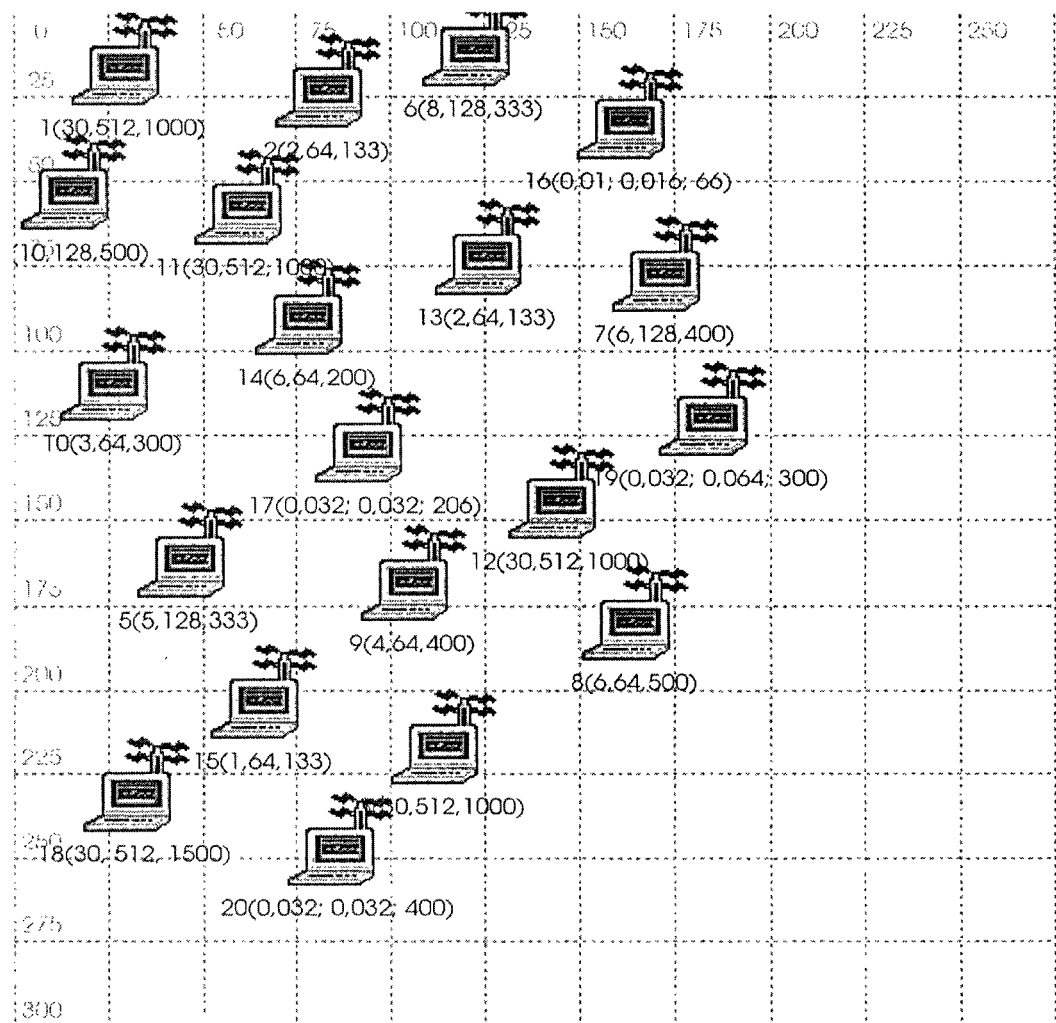


Figure 4.2 Configuration du réseau de test

Nous avons également considéré (avec une portée de transmission fixée à 45 m) le cas où l'algorithme WCA se comporte comme l'algorithme "highest degree heuristic". Rappelons que cette heuristique considère comme tête de grappe, l'unité mobile qui a le plus de voisins, ce qui correspond (formule 2.6) à $w_2 = w_3 = w_4 = 0$. Nous utilisons la configuration de la Figure 4.2 où seuls les éléments numérotés de 1 à 15 sont considérés. Le Tableau 4.3 montre les résultats. Les têtes de grappe choisies sont dans la première colonne de ce tableau et les éléments de la seconde colonne sont les membres des grappes formées.

En analysant les éléments choisis et leur degré au Tableau 4.2, nous remarquons que les résultats obtenus répondent aux attentes de cet algorithme. En effet, l'assignation des identificateurs (ID) a été basée sur la mobilité pour considérer le cas où le WCA se comporte comme l'algorithme "Lowest-Id heuristic". Le plus petit ID est attribué à l'unité mobile la moins rapide et le plus grand à l'unité mobile la plus rapide. Dans ce cas, $w_1 = w_2 = w_4 = 0$ et $w_3 = 1$. Ces tests nous permettent de valider l'implémentation faite. Dans nos expériences, trois configurations de 12, 15 et 20 unités mobiles sont considérées.

Pour valider l'implémentation faite, nous avons analysé, sur la configuration de la Figure 4.2, les unités mobiles choisies par notre algorithme. Ce dernier est baptisé SRCAC (Stability and Resource Consideration Algorithm for Clustering in Mobile Ad hoc Networks). La portée de transmission est fixée à 95 m, le coefficient d'indice de potentialité à 0.95 et celui de stabilité à 0.05. Les indices de potentialité, de stabilité et d'élection de ces éléments sont représentés au Tableau 4.5. Les unités mobiles 18, 11, 12, 16 sont choisies comme têtes de grappe. Cela est conforme à nos attentes. En effet, les unités mobiles 18, 12 et 11 sont les plus pourvues en ressources et procurent une stabilité aux grappes formées. Les membres de la grappe formée par 16 sont 7, 6 et 19. Les éléments 7 et 6 qui ont des indices d'élection (voir le Tableau 4.5) plus grands que 16 sont des passerelles, ce qui ne contredit pas notre proposition. Les autres grappes sont

18 (3,15,2,5,20), 12 (3,2,9,7,8,13,14) et 11 (1,4,6,13,14,10,17) où les unités mobiles entre parenthèses sont membres. Les résultats obtenus sont donc conformes à nos attentes.

Nous allons maintenant passer à une série de tests pour comparer les deux algorithmes. À partir d'une configuration initiale choisie au hasard, nous déplaçons uniformément les éléments dans des directions aléatoires. À chaque fois, nous varions la portée de transmission de 10 m, et nous analysons le nombre de têtes de grappe dans le réseau. Nous traitons le cas d'un réseau de 12, 15, 20. Nous supposons des valeurs du Seuil_min et du Seuil_max respectivement de 4 et 10. Dans l'algorithme WCA, nous avons maintenu les valeurs des coefficients telles que présentées dans (Chatterjee et al., 2002): $w_1 = 0.7$, $w_2 = 0.2$, $w_3 = w_4 = 0.05$.

Tableau 4.2 La mobilité et le degré de chacun des 15 éléments du réseau

UM	Mobilité	Degré
1	14.0769	3
2	11.3967	2
3	40.4877	6
4	15.4961	4
5	30.3008	3
6	27.9224	6
7	26.0823	5
8	11.7942	3
9	51.1483	3
10	36.4451	6
11	16.7222	5
12	51.6054	4
13	33.5209	3
14	19.5456	5
15	15.3103	5

La Figure 4.2 illustre la topologie du réseau à 20 éléments. Dans cette figure, nous associons à chaque unité mobile son identificateur (Id), sa capacité ROM, sa capacité RAM ainsi que la vitesse de son processeur. Les Figures 4.2, 4.3 et 4.4

montrent les résultats. Le SRCAC (stability and ressources consideration algorithm in clustering ad hoc network) est le nom donné à notre algorithme.

La même expérience a été faite en partant de la configuration de la Figure 4.2 mais avec des unités mobiles ayant des portées de transmission différentes. Cela est aussi possible car les éléments du réseau peuvent avoir des interfaces de communication de types différents. Les résultats obtenus sont présentés à la Figure 4.5.

Tableau 4.3 Comportement du “WCA” en “highest degree heuristic”

TDG	UM affiliée à la grappe
3	6, 10, 7, 12, 13
11	14, 2, 4, 1
15	8
5	-
9	-

Tableau 4.4 Comportement du “WCA” en “Lowest-Id heuristic”

TDG	UM affiliée à la grappe
2	8, 15, 11
1	4
14	6, 5
7	13, 3
10	9, 12

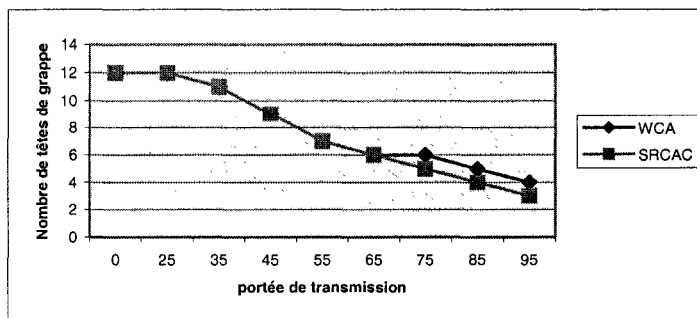
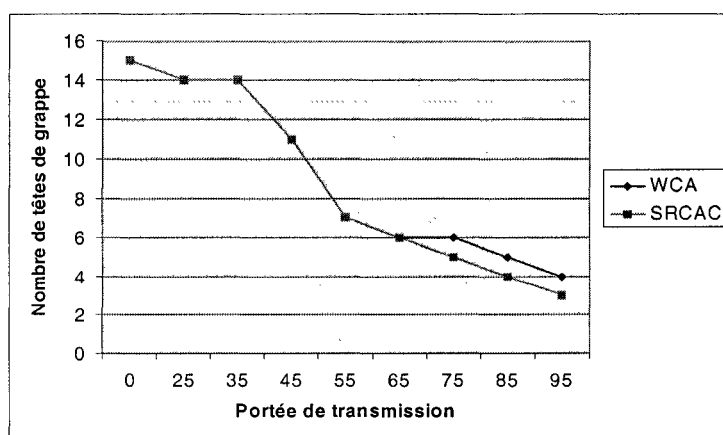


Figure 4.3 Comparaison des deux algorithmes sur un réseau de 12 éléments

Tableau 4.5 Les indices de potentialité, de stabilité et d'élection

ID	indice de potentialité	indice de stabilité	Indice d'élection
18	159.6	1602.13	231.727
3	159.6	1031.81	203.21
12	159.6	126.706	157.955
11	159.6	3.37093	151.789
1	159.6	0.089523	151.624
15	19.4	1249.34	80.8971
2	19.6	959.969	66.6184
9	20	724.44	55.222
7	39.6	283.323	51.7861
5	39.4	71.7893	41.0195
4	40.4	9.38505	38.8493
6	40	1.0769	38.0538
8	20.4	269.722	32.8661
13	19.6	135.918	25.4159
14	20.4	0.759915	19.418
10	19.8	6.76922	19.1485
16	0.0068	98.4492	4.92892
20	0.016	37.6207	1.89623
19	0.0256	0.256975	0.0371688
17	0.016	0.0197425	0.0161871

**Figure 4.4 Comparaison des deux algorithmes sur un réseau de 15 éléments**

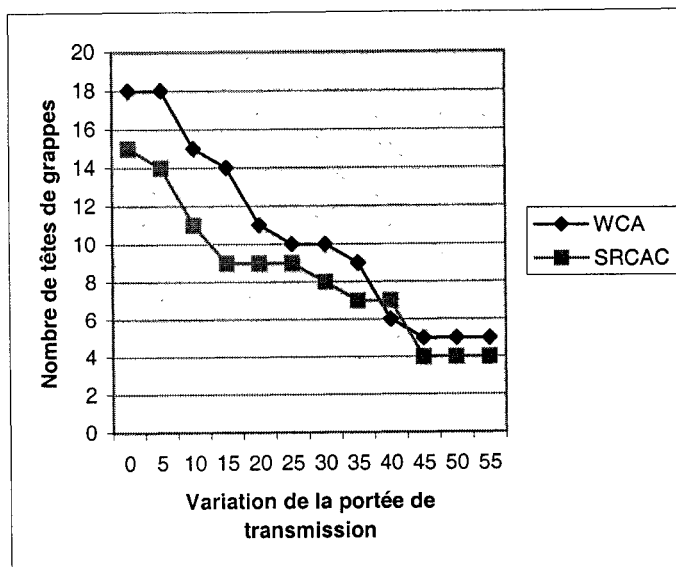


Figure 4.5 Nombre de têtes de grappes pour un réseau de 20 éléments ayant des portées de transmission différentes

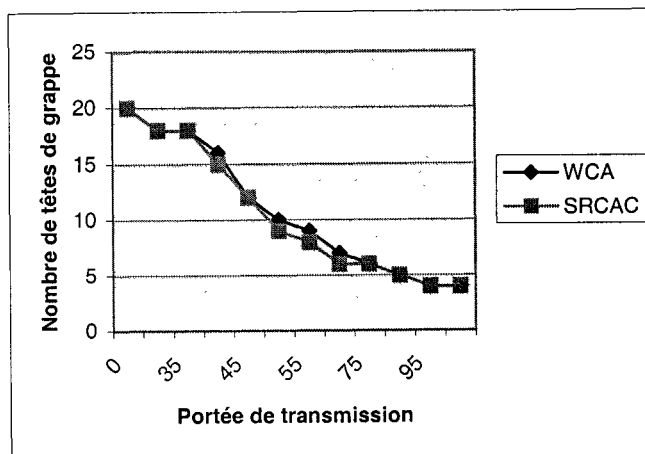


Figure 4.6 Comparaison sur un réseau de 20 éléments ayant une même portée de transmission

Nous remarquons que pour les deux algorithmes, le nombre de têtes de grappes diminue avec l'augmentation de la portée de transmission. Cela s'explique par le fait qu'une tête de grappe qui a une portée de transmission élevée couvre une grande zone (donc un grand nombre d'unités mobiles).

En comparant les deux algorithmes sur la base de la métrique du nombre de têtes de grappe, nous pouvons affirmer que l'algorithme proposé (SRCAC) donne une amélioration de 4.9% par rapport au WCA. Cela est dû au fait que l'indice de stabilité qui est la partie variable de l'indice d'élection varie avec la portée de transmission. Cette dernière est prise en considération dans le choix d'une tête de grappe par le SRCAC seulement. En outre, l'avantage de notre proposition est que nous sommes rassurés qu'à chaque fois, les unités mobiles choisies sont capables d'assurer ce rôle et les durées de vie des grappes formées sont relativement élevées. Le stockage des données de la grappe, les tables de routage ainsi que le traitement des requêtes nécessitent des ressources qu'on peut plus facilement trouver dans un Laptop que dans un simple PDA. En effet, pour une portée de 100 m et un même nombre de grappes (4), le réseau de la Figure 4.2 forme les grappes du Tableau 4.6

Tableau 4.6 Grappes formées à partir du réseau de la Figure 4.2

WCA		Algorithme Proposé	
TDG	UM Affiliées	TDG	UM Affiliées
16	6, 19, 7, 13	18	3, 15, 2, 9, 5, 20
4	1, 11, 10, 17, 14	12	3, 15, 2, 9, 8, 7, 13
8	3, 2, 12, 9	11	1, 13, 4, 6, 10, 14, 17
20	18, 15, 5	16	7, 6, 19

En analysant ces résultats, nous voyons d'une part que l'unité mobile 16 est choisie par les deux algorithmes. D'autre part, les unités mobiles 18, 12, 11 sont les plus pourvues en ressources, tandis que 4, 8, 20 sont les moins pourvues en ressources (à constater à la Figure 4.2), ce qui montre l'importance du SRCAC. Dans l'algorithme WCA, si par exemple l'UM 11 (30, 512, 1000) veut communiquer avec l'UM 18 (30, 512, 1000), elle est obligée de passer par les têtes de grappe 4 (10, 128, 500) et 20 (0.032, 0.032, 400). Ces dernières peuvent ne pas être en mesure d'assurer la tâche

convenablement. En outre, nous n'avons pas considéré le cas où l'énergie serait un facteur limitant, nous supposons que les unités mobiles sont dotées de batteries rechargeables (comme les batteries à combustibles).

4.4 Implémentation du protocole ODMRP avec prédiction de mobilité

La seconde partie de notre implémentation vise à intégrer les contraintes de QoS (Qualité de Service) au protocole ODMRP avec prédiction de la mobilité. Nous utilisons la simulation pour évaluer les performances de nos propositions. Des outils de simulation comme NS (*Network Simulator*) (McCanne et Floyd, 2000), OPNET (Akhtar, 1997) et GloMoSim (*Global Mobile Information System Simulator*) (Zeng et al., 1998) peuvent être utilisés. Nous avons choisi d'utiliser GloMoSim car il intègre déjà la version du protocole ODMRP que nous voulons améliorer. Dans ce qui suit, nous présentons le simulateur GloMoSim, l'implémentation du protocole ODMRP avec prédiction de la mobilité ainsi que nos ajouts. Dans la suite, nous présentons les paramètres d'évaluation des performances, le plan de test ainsi que les résultats.

4.4.1 Bref aperçu de GloMoSim

GloMoSim est un simulateur de réseau utilisé pour évaluer les performances des réseaux sans fil. Il utilise une simulation parallèle à événements discrets fournie par Parsec (*Parallel Simulation Environment for Complex System*) (Bagrodia et al., 2000) développé par PCL (*Parallel Computing Laboratory*) de l'UCLA (*University of California at Los Angeles*).

GloMosim a une structure de couches semblable à celle du modèle OSI (Open Systems Interconnection). En effet, chaque entité intègre les diverses couches réseaux qui peuvent être simulées sous forme de fonction. Au début de la simulation, une fonction d'initialisation est appelée pour chaque couche. Dès qu'une couche reçoit un message, elle exécute une tâche demandée par cette dernière. À la fin de la simulation, une nouvelle fonction est lancée pour mettre fin à la simulation et collecter les statistiques désirées.

GloMoSim contient plusieurs répertoires à savoir :

- */application* : ce répertoire contient le code pour la couche application ;
- */bin* : il contient les fichiers exécutables ainsi que les fichiers d'entrée et de sortie ;
- */doc* : contient la documentation ;
- */include* : contient les fichiers communs .

Les répertoires */mac*, */network*, */radio*, */transport* contiennent respectivement les codes pour les couches MAC, réseau, physique et transport.

La simulation se fait sur une station windows. La compilation génère dans le répertoire */bin*, un exécutable nommé *glomosim.exe*. Dans la simulation, GloMosim prend en entrée un fichier de configuration nommé *config.in* et les statistiques sont collectées dans un autre fichier nommé *Glomo.stat*.

Le fichier de configuration contient plusieurs variables (ou paramètres) de la simulation parmi lesquelles nous retrouvons :

- Temps de simulation (*Simulation-Time*): définit le temps de simulation (en jour, heure, seconde, micro-seconde et nano-seconde) ;
- Dimensions du terrain (*Terrain-Range-X* et *Terrain-Range-Y*) : montrent la largeur et la longueur du terrain de simulation (en mètre) ;
- Partition-Num-X et Partition-Num-Y : définissent le nombre de partitions sur les axes des abscisses et des ordonnées du terrain de simulation ;
- Nombre de nœuds (*Number-of-Nodes*) : indique le nombre d'unités mobiles dans la simulation ;
- Application : indique l'application utilisée dans la simulation (FTP, HTP, CBR ou TELNET) ;
- Protocoles de routage, de transport et de la couche MAC (Routing-Protocol, Transport-Protocol, MAC-Protocol) : indiquent respectivement les protocoles de routage, de transport utilisé (UDP ou TCP) et de la couche Mac (802.11, CSMA, FAMA : *Floor Acquisition Multiple Access*, MACA : *Multiple Access Collision Avoidance*) ;

- Mobilité (*Mobility*) : spécifie si les unités mobiles sont stationnaires ou non. Au cas où la mobilité est activée, on spécifie alors sa nature (aléatoire ou pas), la vitesse de déplacement minimale (*Mobility-Wp-Min-Speed*) et maximale (*Mobility-Wp-Max-Speed*), le temps de pause (*Mobility-Wp-Pause*) ainsi que la trajectoire suivie (*Mobility-Trace* et *Mobility-Trace-File*).

GloMoSim supporte deux approches de mobilité, à savoir le « *Random Waypoint* » et le « *Random Druken* ».

1. « *Random Waypoint* » : c'est une approche dans laquelle la destination est choisie aléatoirement dans le terrain de simulation. L'utilisateur spécifie dès le départ deux valeurs minimales et maximales de la vitesse et, lors du déplacement, l'unité mobile choisit aléatoirement une vitesse comprise dans ses limites.
2. « *Random Druken* » : dans cette approche, une unité mobile se déplace périodiquement, à une vitesse donnée, d'une des positions de ces voisins immédiats à une position arbitraire. En effet, si elle se trouve à la position (x, y) , elle peut se déplacer aux positions $(x-1, y)$, $(x+1, y)$, $(x, y-1)$ ou $(x, y+1)$.

4.4.2 ODMRP avec prédiction de la mobilité

Pour analyser l'effet de nos propositions, nous avons d'abord implémenté la prédiction de la mobilité telle que spécifiée dans (Lee et al., 1999). En effet, nous n'avons pas pu trouver une version de GloMoSim qui intègre cette méthode. La version utilisée est GloMoSim 2.0. Nous avons ajouté des champs supplémentaires dans les entêtes des paquets *Message* (le paquet est utilisé pour établir une route entre une source et une destination) et *Join_Reply*. Ce dernier est envoyé par la destination pour signifier à la source la route choisie.

Quand une source envoie une requête de construction de route (paquet *Message*), elle applique ses coordonnées, sa vitesse ainsi que sa direction de déplacement. En outre, elle initialise un champ destiné à conserver le temps d'expiration d'une route (RET) à une valeur élevée nommée MAX-LET-VALUE. Une unité mobile qui reçoit la requête de construction de route (*Message*) prédit le temps d'expiration du lien (LET) entre elle

et son prédécesseur. Si l'unité mobile n'est pas un membre du groupe *multicast*, elle remplace les paramètres de localisation et de mobilité du paquet *Message* par les siennes. Le minimum entre la valeur du LET calculée et sa valeur courante est affectée à la variable *Minimum Link Expiration Time*. Si l'unité mobile est un récepteur *multicast*, le minimum entre le LET calculé et celui spécifié dans la requête de construction de route est considéré comme RET (Route Expiration Time). Elle génère par la suite un paquet *Join_Reply* dans lequel elle insère le RET. Le *Join Reply* est alors diffusé. Si un membre du groupe de multi-diffusion reçoit plusieurs tables *Join_Reply*, il insère la plus petite valeur du RET dans une table unique qu'il rediffuse par la suite.

4.4.3 ODMRP avec prédiction de la mobilité et contraintes de qualité de service

Pour intégrer les contraintes de qualité de service, il a également fallu modifier les paquets *Message* et *Join_Reply*. Dans cette section, nous montrons les modifications faites aux paquets utilisés dans l'établissement d'une route.

Quand une source envoie une requête de construction de route, elle y ajoute ses exigences de qualité de service (*QOS_BP_exige*, *QOS_delai_exige*, *QOS_TC_exige*). Elle ajoute également deux champs pour conserver respectivement la bande passante minimale et la capacité minimale du tampon sur tout le chemin. La source initialise ces champs aux valeurs initiales des deux paramètres.

Deux options peuvent être utilisées pour implémenter notre proposition. Dans la première, l'unité mobile qui reçoit la requête vérifie si la bande passante et le tampon exigés sont inférieurs à la bande passante et le tampon dont elle dispose. Si oui, elle diffuse la requête sur tous les liens qui satisfont à ces contraintes. Cette façon d'implémenter peut introduire une complexité énorme si le nombre de sources augmente. Dans la seconde, l'unité mobile qui reçoit le paquet *Message* remplace les valeurs des champs *Minimum B_P* et *Minimum T_C* par les minima entre ces valeurs et celles de la bande passante et tampon disponibles avant de diffuser le paquet. Le processus continue jusqu'à la tête de grappe qui héberge la destination. Cette tête de grappe sélectionne la route la plus stable parmi celles qui satisfont aux contraintes. Elle

génère par la suite un paquet *Join-Reply* dans lequel elle indique la bande passante et la taille du tampon réservées. Toutes les unités mobiles sur la route sélectionnée font une mise à jour des ressources disponibles.

Étant donné la complexité qu'impose la première option, la seconde a été choisie pour implémenter nos propositions. Ainsi, la structure *glom_node_str* (qui décrit une unité mobile dans le réseau) a été modifiée pour tenir compte du statut, des ressources (avant et après réservation) de chaque élément du réseau ainsi que les spécifications en qualité de service.

4.5 Évaluation des performances du protocole ODMRP

Dans cette section, nous décrivons l'environnement de simulation et nous interprétons les résultats. Nous avons utilisé un réseau de 20 unités mobiles placées suivant la configuration de la Figure 4.2. Les positions exactes de ces unités mobiles sont spécifiées dans le fichier *nodes.input*. Dans la version originale de ODMRP (issue de GloMoSim 2.0), seules les coordonnées sont indiquées. Pour analyser l'impact de nos propositions, nous avons dû modifier ce fichier en ajoutant le statut, les ressources initiales (bande passante initiale, tampon initial) ainsi que des variables qui spécifient les exigences de qualité de service. La mobilité est spécifiée dans le fichier *mobility.in*. Nous considérons une portée de transmission de 80 m, une bande passante de 2 Mbits/sec, un modèle de propagation en espace libre et nous utilisons le 802.11 comme protocole de la couche MAC.

Pour analyser les performances de nos propositions, nous évaluons l'impact des contraintes de qualité de service sur les performances du protocole ODMRP avec prédiction de la mobilité. Rappelons que, dans les deux cas, la métrique de choix de route reste le RET, sauf que dans notre proposition la route doit préalablement satisfaire aux contraintes imposées. Deux variables sont susceptibles de changer : il s'agit du délai de bout-en-bout (end-to-end delay) et de la surcharge de contrôle (control overhead). Nous gardons l'application (Constant Bite Rate : CBR) considérée dans (Su et al., 2001).

Nous varions deux paramètres, à savoir la densité du groupe *multicast* ainsi que le nombre de sources *multicast*. Dans un premier scénario, nous maintenons le nombre de sources *multicast* à 2 et nous varions la densité du groupe *multicast* de 2 à 10. Dans un second scénario, la taille du groupe *multicast* est maintenue à 10 et nous varions le nombre de sources *multicast* de 1 à 10. La Figure 4.6 montre la variation du délai de bout-en-bout en fonction de la taille du groupe *multicast*. La Figure 4.7 illustre la surcharge de contrôle en fonction du nombre de sources.

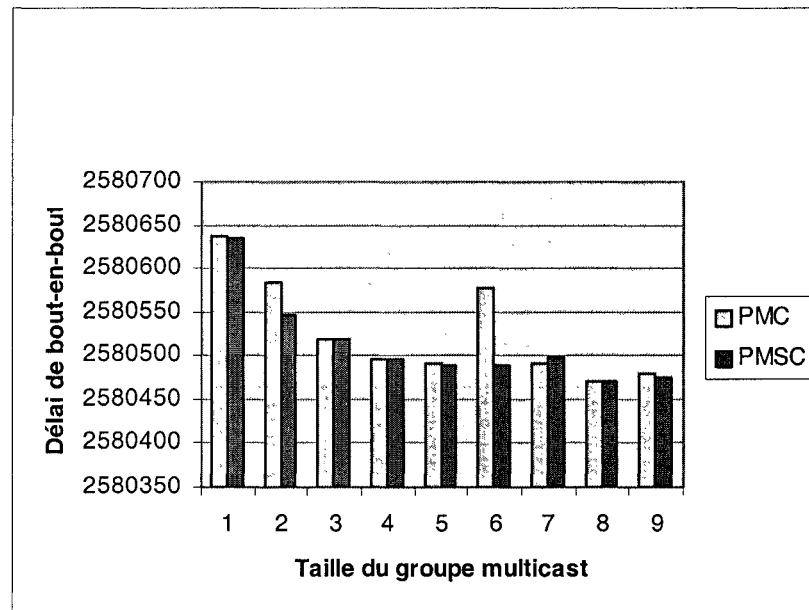


Figure 4.7 Délai de bout-en-bout en fonction de la taille du groupe *multicast*

En analysant les résultats de la Figure 4.6, nous remarquons que le délai de bout-en-bout diminue avec la taille du groupe *multicast*. Les délais sont presque les mêmes pour les deux schémas PMC (prédiction de la mobilité avec contraintes) et PMSC (prédiction de la mobilité sans contraintes). La petite différence (insignifiante) est due au temps d'exécution pendant lequel la tête de grappe qui héberge la destination vérifie les contraintes de qualité de service. La Figure 4.7 montre la variation de la surcharge de contrôle (contrôle overhead) en fonction du nombre de sources. La surcharge de contrôle augmente avec le nombre de sources *multicast* dans les deux schémas (PMC et PMSC).

Nous voyons que, dans notre scénario, nous avons les mêmes valeurs sauf dans le cas de 3 sources et de 6 sources.

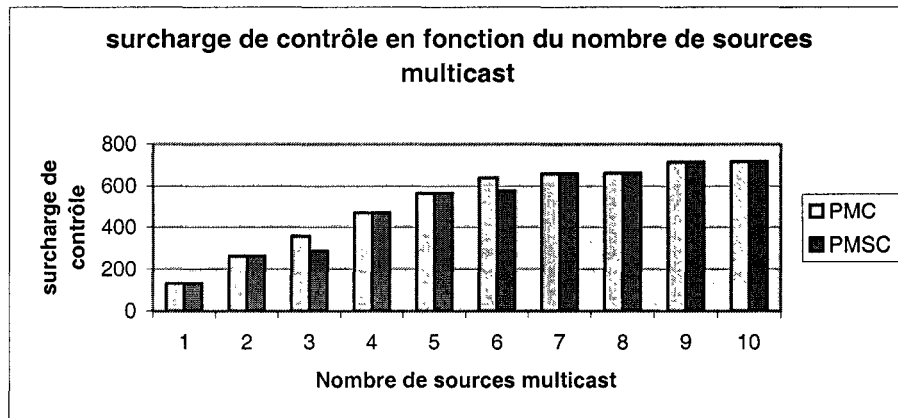


Figure 4.8 surcharge de contrôle en fonction du nombre de sources

Ainsi, nous avons présenté les contraintes qu'il faudrait intégrer dans le choix d'une route stable. En ajoutant ces contraintes, le protocole ODMRP avec prédiction de la mobilité assure que la route choisie sera capable d'acheminer du trafic ayant des exigences de qualité de service. La détérioration des performances (délai de bout-en-bout et la surcharge de contrôle) imposée par l'ajout de telles contraintes est insignifiante.

CHAPITRE V

CONCLUSION

5.1 Synthèse des travaux

Les caractéristiques des réseaux mobiles ad hoc font qu'ils utilisent, dans le routage, des mécanismes de diffusion alors que l'un des défis dans les réseaux sans fil est l'optimisation de la bande passante. Les protocoles recensés dans la littérature se regroupent en deux grandes catégories: les protocoles proactifs et les protocoles réactifs. Avec l'apparition du système de positionnement GPS (Global Positioning System), des protocoles basés sur la localisation ont vu le jour. Une méthode de prédiction de la mobilité a été introduite pour améliorer ces protocoles. Nous nous sommes servis du protocole ODMRP avec prédiction de la mobilité. En effet, ce dernier a été identifié dans la littérature comme étant un des plus robustes parmi les protocoles *multicast*. Dans ce protocole, la route est choisie en utilisant comme métrique la stabilité (matérialisée par le RET). Cependant, une route plus stable peut ne pas répondre au critère de délai exigé par les applications temps réel. En outre, la méthode n'a pas prévu des contraintes sur la capacité d'un lien ou celle du tampon au niveau des unités mobiles de transit. Notre contribution par rapport à cela a été d'intégrer les contraintes de qualité de service dans le choix d'une route stable.

Dans le processus de découverte et de maintenance de route, les unités mobiles diffusent et rediffusent des messages de signalisation. Ces messages consomment les ressources réseaux déjà insuffisantes, en l'occurrence la bande passante et l'énergie. Pour optimiser ces ressources, une des solutions consiste à réduire le nombre de participants dans le mécanisme de diffusion en formant des grappes et en choisissant dans chaque grappe des têtes de grappe. La solution que nous proposons et qui constitue notre contribution est de choisir comme participants au mécanisme de diffusion, des unités mobiles dont les ressources et les caractéristiques du mouvement leur permettent de former des grappes stables.

Dans ce mémoire, nous avons présenté un algorithme qui permet d'élire comme tête de grappe des unités mobiles dont l'indice d'élection (une combinaison linéaire de l'indice de stabilité et de l'indice de potentialité) est le plus grand. Nous l'avons comparé à un algorithme nommé WCA. L'implémentation des deux algorithmes s'est fait à l'aide du langage C++. Les unités mobiles et leurs attributs sont stockés dans un fichier. En fixant le nombre d'unités mobiles dans le réseau, nous avons fait varier la portée de transmission et nous avons déterminé le nombre de têtes de grappe (et donc le nombre de grappes) dans le réseau. Les résultats comparatifs ont été présentés sous forme de courbes. En analysant les résultats, nous avons observé que les unités mobiles élues sont les plus pourvues en ressources et ont assuré une meilleure stabilité aux grappes qu'ils forment. En outre, en considérant comme métrique le nombre de têtes de grappe, l'algorithme proposé donne de meilleurs résultats. Cela est dû au fait que l'indice de stabilité qui est la partie variable de l'indice d'élection est une fonction de la portée de transmission. Cette dernière est utilisée pour construire les grappes dans les deux algorithmes mais, dans le « WCA », la portée de transmission a une moindre importance dans le choix d'une tête de grappe.

Dans la seconde partie de ce travail, nous avons proposé des améliorations à apporter à une architecture de base. Ainsi, nous avons construit une maille entre toutes les passerelles qui sont dans la même zone de portée de transmission pour améliorer la fiabilité. Egalement, nous avons proposé une réplication entre une tête de grappe primaire et une tête de grappe secondaire afin d'améliorer la disponibilité du service de l'architecture qui en résulte. Parmi les protocoles de routage *multicast*, ODMRP a été identifié comme étant le plus adapté aux réseaux mobiles ad hoc. Nous avons intégré à une version de ce protocole avec prédiction de la mobilité, les contraintes de qualité de service (QoS). Le simulateur GloMoSim a été utilisé pour évaluer l'impact de nos propositions sur les performances du protocole.

Le délai de bout-en-bout (end-to-end delay) et la surcharge de contrôle (control overhead) étaient les métriques qui pouvaient varier. Cependant, les résultats montrent

que l'impact est insignifiant. Nous pouvons ainsi conclure qu'il faut ajouter les contraintes de qualité de service dans le choix d'une route stable.

5.2 Limitations des travaux

En dépit des résultats obtenus, nos propositions ont été soumises à quelques hypothèses. En effet, dans le réseau ad hoc dense considéré :

- les unités mobiles sont hétérogènes (des laptops avec processeurs, capacités ROM et RAM différentes, des PDA et des téléphones portables avec l'option d'échanges de données). Cependant, notre indice de potentialité perd son intérêt dans le cas où les éléments du réseau ont les mêmes caractéristiques.
- Les unités mobiles sont à l'air libre avec disponibilité du système GPS. Nous n'avons pas considéré le cas contraire. En outre, le dispositif GPS ajoute un poids supplémentaire.
- Les unités mobiles disposent de radios émetteurs-récepteurs équipées de plusieurs éléments d'antennes directionnelles avec possibilité d'activer ou de désactiver une direction relative à une antenne. Nous n'avons pas implémenté cette alternative, nous la réservons pour les travaux futurs.
- Les unités mobiles disposent de sources d'énergie durable, en l'occurrence la pile à combustible. C'est bien une innovation technologique mais elle n'est pas encore très répandue sur le marché.
- Les liaisons radio sont fiables (pas d'obstacles, évanouissements ou interférences) et ont un rapport signal à bruit (S/N) acceptable pour toutes les unités mobiles en communication directe. Toute autre communication dont le rapport S/N est inférieur au seuil est considérée comme inexistante.

Lors de l'initialisation, les unités mobiles diffusent leurs caractéristiques de mouvement. Mais, comme elles sont mobiles, ses dernières peuvent changer pendant le temps de transmission. Il s'ensuit que les informations enregistrées peuvent ne pas être correctes. La supposition que les unités mobiles sont fixes pendant le déroulement de l'algorithme est très simplificatrice. En outre, nous avons supposé que chaque unité

mobile calcule son indice de potentialité avant de le diffuser. Cela suppose que, pour chaque unité mobile, il y a un fichier qui récupère les caractéristiques physiques de chaque machine et qu'il y ait un processus de calcul de l'indice d'élection. Nous supposons dans nos tests que tous les messages transmis arrivent à destination, ce qui n'est pas nécessairement correct car un lien de communication peut se briser, ou l'une des unités mobiles peut s'éteindre ou sortir de la zone de portée de transmission des autres. Nous n'avons pas implémenté les mécanismes de mise à jour et de communication entre les têtes de grappe principale et secondaire. En effet, implémenter toutes les propositions est un travail qui pourrait prendre plus de temps que la durée réglementaire d'une maîtrise.

5.3 Travaux futurs

Nos tests ont été faits à l'aide de la simulation. Dans la comparaison des algorithmes WCA et SRCAC, nous avons utilisé le nombre de têtes de grappe comme métrique. Dans le futur, il faudrait considérer les deux autres métriques à savoir le nombre de ré-affiliations et le nombre de mises à jour. Nous pouvons également considérer un cas réel et essayer de vérifier nos propositions. Il est également souhaitable d'implémenter le cas où les unités mobiles disposeraient des interfaces de communications différentes (présence des antennes omnidirectionnelles et directionnelles) ainsi que les différentes mises à jour entre les têtes de grappe principale et secondaire. On peut également considérer le cas où les éléments du réseau ne se trouvent pas à l'air libre et analyser l'impact du milieu de test sur les performances des différentes versions du protocole.

BIBLIOGRAPHIE

Akhtar H., “An Overview of Some Network Modeling, Simulation, Performance Analysis Tools,” *Proceedings of 2nd IEEE Symposium on Computers and Communications*, 1997.

Baker D.J. et Ephremides A., “A distributed algorithm for organizing mobile radio telecommunication networks,” *the Proceedings of the 2nd International Conference on Distributed Computer Systems*, pp. 476–483, Apr. 1981.

Baker D.J. et Ephremides A., “The architectural organization of a mobile radio network via a distributed algorithm,” *IEEE Transactions on Communications COM-29*, pp. 1694–1701, Nov. 1981.

Banâtre M., Couderc P., Menaud J. et Weis F., “Proximate Interactions for Wireless Appliances,” *Proceedings of the 9th ACM SIGOPS European Workshop - Beyond the PC: New Challenges for the Operating System*, Kolding, Denmark, pp. 127 – 132, Sep. 2000.

Basagni S., “Distributed clustering for ad hoc networks,” *Proceedings of the IEEE International Symposium on Parallel Architectures, Algorithms and Networks*, Western Australia, pp. 310–315, Jun. 1999.

Basagni S., Chlamtac I. et Farago A., “A generalized clustering algorithm for peer-to-peer networks,” *Proceedings of Workshop on Algorithmic Aspects of Communication (satellite workshop of ICALP)*, Jul. 1997.

Basagni S., Chlamtac I., Syrotiuk V. R. et Woodward B. A. “A distance routing effect algorithm for mobility (DREAM),” *Proceeding of ACM/IEEE MOBILECOM'98*, Dallas,

Texas, pp. 76-84, Oct. 1998.

Bommaiah E., Liu M., McAuley A. et Talpade R., "AMRoute: Ad Hoc Multicast Routing Protocol," *Internet-Draft, draft-talpade-manet-amroute.txt*, Work in progress, Aug. 1998.

Capkun S., Hamdi M. et Hubaux J. P., "GPS-free positioning in Mobile Ad Hoc Networks," *Cluster Computing*, vol. 5, pp. 157-167, 2002.

Charles E.P. et Bhagwat P., "Highly Dynamic Destination-Sequenced Distance Vector Routing (DSDV) for Mobile Computers," *Proceedings of the ACM SIGCOMM Conference on Communications Architectures, Protocols and Application*, London, UK, pp. 234-244, Sep. 1994.

Charles E.P. et Elisabeth M. R., "Ad hoc on demand distance vector (AODV) algorithm," *Proceeding of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99)*, New Orleans, Louisiana, USA, vol. 3, pp. 90-100, Feb. 1999.

Chatterjee M., Das S.K. et Turgut D., "WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks," *Cluster Computing*, vol. 5, pp. 193-204, 2002.

Chen T.-W et Gerla M., "Global State Routing: A new routing scheme for ad-hoc wireless networks," *Proceeding IEEE ICC'98*, pp. 171-175, Jun.1998.

Chiang C.-C, Gerla M. et Zhang L., "Shared Tree Wireless Network Multicast," *Proceedings of IEEE IC3N '97*, Las Vegas, NV, Sep. 22 - 25, 1997.

Chiang C.-C, Gerla M. et Zhang L, "Forwarding Group Multicast Protocol (FGMP) for Multihop, Mobile Wireless Networks," *ACM-Baltzer Journal of Cluster Computing: Special Issue on Mobile Computing*, vol. 1, no. 2, pp.187-196, Dec. 1998.

Corson S., Macker J., "rfc2501 Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," *University of Maryland, Naval Research Laboratory*, Jan. 1999.

Felber P. et Schiper A., "Optimistic active replication," *Proceedings 21st International Conference on Distributed Computing Systems (ICDCS'2001)*, Phoenix, AZ. IEEE Computer Society, 2001.

Garcia-Luna-Aceves J.J. et Spohn M., "Scalable Link-State Internet Routing," *Proceedings Sixth International Conference on Network Protocols*, pp. 52–61. Oct. 1998.

Garcia-Luna-Aceves J.J. et Spohn M., "Source-Tree Routing in Wireless Networks," *Proceedings IEEE ICNP 99: 7th International Conference on Network Protocols*, Toronto, Canada, Oct. 31_Nov. 3, 1999.

Garcia-Luna-Aceves J.J et Madruga E. L., "The Core Assisted Mesh Protocol (CAMP)," *IEEE Journal on Selected Areas in Communications, Special Issue on Ad-Hoc Networks*, Vol. 17, no. 8, pp. 1380-1394, Aug.1999.

Gerla M. et Tsai J.T.C., "Multi cluster, mobile, multimedia radio network, Wireless Networks," *ACM-Baltzer Journals*, vol. 1, no. 3, pp. 255–265, Jul. 1995.

Haas Z. J. et Pearlman M., "The Zone Routing Protocol (ZRP) for ad hoc networks", *Technical report, Internet Draft*, Nov. 1997.

Iwata A., Chiang C.-C., Pei G., Gerla M. et Chen T.-W., "Scalable Routing Strategies for Ad Hoc Wireless Networks," *IEEE Journal on Selected Areas in Communications, Special Issue on Ad-Hoc Networks*, pp. 1369-79, Aug. 1999.

Jiang M., Li J. et Tay Y., "Cluster Based Routing Protocol (CBRP) Functional Specification," *Internet Draft, draft-ietf-manet-cbrp.txt*, work in progress, Jun. 1999.

Johnson D.B. et Maltz D. A., "Dynamic source routing in ad hoc wireless networks," *Mobile Computing, Mc Kluwer Academic Publishers*, vol. 353, pp. 153-181, Mar. 1996.

Kaplan E.D. (Editor), *Understanding the GPS: Principles and Application*, Artech House, Boston, MA, Feb. 1996.

Kwon T. J. et Gerla M., "Efficient Flooding with Passive Clustering (PC) in Ad Hoc Networks", *ACM Computer Communication Review*, vol. 32, no. 1, pp. 44-56, Jan. 2002.

Lee S.-J., Su W. et Gerla M., "Ad hoc wireless multicast with mobility prediction", *Proceedings of IEEE ICCCN'99*, Boston, MA, pp. 4-9, Oct. 1999.

Lee S.-J., Su W., Gerla M. et Bagrodia R., "A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols," *Proceedings of IEEE INFOCOM 2000*, Tel Aviv, Israel, pp. 565-574, Mar. 2000.

Lee S.-J., Gerla M., et Chiang C.-C., "On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks On-Demand Multicast in Mobile Wireless Networks," *Proceedings of IEEE WCNC'99*, New Orleans, LA, pp. 1298-1302, Sep. 1999.

Lin C.R. et Gerla M., "Adaptive Clustering for mobile wireless networks," *IEEE Journal on selected Areas in Communications*, vol. 15, no. 7, pp. 1265-1275, Sep. 1997.

McDonald A.B., "A mobility-based framework for adaptive clustering in wireless ad hoc network," *IEEE Journal on Selected Areas in communications*, vol. 17, no. 8, pp.1466-87, Aug. 1999.

Murthy S. et Garcia-Luna-Aceves J. J., "An Efficient Routing Protocol for Wireless Networks," *ACM Mobile Networks and Applications Journal*, Special Issue on Routing in Mobile, Communication Networks, pp. 183-97, Oct. 1996.

Murthy S. et Garcia-Luna-Aceves J. J., "A routing protocol for packet radio networks," *Proceeding of the IEEE Mobicom*, pp. 86-95, Nov. 1995.

Park V. D. et Corson M.S., "A highly adaptive distributed routing algorithm for mobile wireless networks," *Proceeding INFOCOM '97*, pp. 1405-1413, Apr. 1997.

Park V. D. et Corson M.S., "Temporally-Ordered Routing Algorithm (TORA) Version 1, functional specification," *IETF, Internet Draft, draft-ietf-manet-tora-spec-02.txt*, Oct. 1999.

Qayyum A., Viennot L. et Laouiti A., "A Multipoint relaying: An efficient technique for flooding in mobile wireless networks," *INRIA research report RR-3898*, 2000.

Royer E.M., "Hierarchical Routing in Ad hoc Mobile Networks," *Wireless Communication & Mobile Computing*, Vol. 2, no. 5, pp. 515-532, 2002.

Royer E.M. et Chai-Keong Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks", *IEEE Personal Communications*, Vol. 6, no. 2, pp. 46-55,

Communication & Mobile Computing, Vol. 2, no. 5, pp. 515-532, 2002.

Royer E.M. et Chai-Keong Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks", *IEEE Personal Communications*, Vol. 6, no. 2, pp. 46-55, Apr. 1999.

Su W., Lee S.J et Gerla M., " Mobility prediction and routing in *ad hoc* wireless networks," *International Journal of Network Management*, Vol. 11, no.1, pp. 3-30, Jan. – Feb. 2001.

Takai M., Lokesh B., Rajat A., Tang K., Bagrodia R. et Gerla M., "GloMoSim: A Scalable Network Simulation Environment," Technical report 990027, UCLA, Computer Science Department, 1999.

Toh C.-K., "Associativity-Based Routing for Ad-Hoc Mobile Networks," *Wireless Personal Communications Journal*, vol. 4, no. 2, pp. 103-139, Mar. 1997.

Vaidya N.H., Krishna P., Chatterjee M. et Pradhan D. K., "A Cluster-based approach for routing in dynamic network," *ACM Computer Communications Review*, vol. 27, no. 2, pp. 372-378, Mar.1997.

Wu C.W. et TAY Y.C., "Ad hoc Multicast Routing protocol utilizing Increasing id-numbers (AMRIS)," *Proceedings of IEEE MILCOM'99*, Atlantic City, NJ, Nov. 1999.

Young-Bae Ko, Shankarkumar V. et Vaidya N. H., "Medium access control protocols using directional antennas in ad hoc networks," *Proceedings of IEEE INFOCOM*, pp. 13-21, Mar. 2000.

Young-Bae Ko et Vaidya N. H., "Location-Aided Routing (LAR) in Mobile Ad Hoc

Networks,” *Proceedings of ACM/IEEE MOBICOM’98*, Dallas, TX, pp. 66-75, Oct. 1998.

Young-Bae Ko et Vaidya N. H, “Location-Aided Routing (LAR) in mobile ad hoc networks,” *Wireless Networks*, vol. 6, no. 4, pp. 307-321, 2000.

Zeng X., Bagrodia R., et Gerla M., “Glomosim: a library for parallel simulation of large-scale wireless networks,” *Proceedings of PADS ’98*, Banff, Canada, Ma. 1998.